The Intelligent Management System for your Facility

User Manual

October 2015
Calpendo 7.0.52

© 2010-2015 Exprodo Software

Table of Contents

Foreword 0

Part I Getting Started 10

Part II Version History 12

2.1 Changes in Version 7.0 ... 12
2.1.1 Changes For Regular Users ... 12
2.1.2 Changes For Administrators ... 13
2.2 Changes In Version 7.0.1 - 7.0.45 ... 17
2.2.1 Changes For Regular Users ... 17
2.2.2 Changes For Administrators ... 18
2.3 Changes in Version 7.0.46-7.0.55 ... 23
2.3.1 Changes For Regular Users ... 23
2.3.2 Changes For Administrators ... 23

Part III Calpendo User Guide 28

3.1 Getting A User Account ... 28
3.2 Change Password ... 31
3.3 User Settings ... 32
3.3.1 Calpendo User Settings ... 32
3.3.2 Date & Time ... 35
3.3.3 Email ... 35
3.3.4 Menu ... 35
3.4 Bookings ... 36
3.4.1 Bookings Calendar ... 36
3.4.1.1 Exploring The Bookings Calendar ... 37
3.4.1.2 Display Of The Bookings Calendar ... 47
3.4.1.3 Creating Bookings ... 49
3.4.1.4 Editing And Cancelling Bookings ... 54
3.4.2 Booking Restrictions ... 57
3.4.2.1 Time Templates ... 57
3.4.2.2 Booking Rules ... 60
3.4.2.3 Permissions ... 60
3.4.3 Searching For Bookings ... 61
3.4.3.1 Using The Booking Searches ... 61
3.4.3.2 Booking Cancellations ... 65
3.4.3.3 My Bookings ... 66
3.4.3.4 My Projects' Bookings ... 67
3.4.3.5 Booking Search ... 68
3.5 Projects ... 69
3.5.1 Creating Projects ... 70
3.5.2 Modifying Projects ... 72
3.5.3 Searching For Projects ... 72
3.5.3.1 Using The Project Searches ... 72
3.5.3.2 My Projects ... 76
3.5.3.3 Project Search ... 77
3.5.4 Project Membership Request ... 78
3.6 Resource Usage ... 79
3.6.1 Resource Usage Session Recorder ... 79
3.6.2 Resource Usage Calendar ... 82

© 2010-2015 Exprodo Software

3.6.3 Searching For Resource Usage ... 84
3.6.3.1 Using the Resource Usage Searches ... 84
3.6.3.2 My Resource Usage ... 87
3.6.3.3 My Projects' Resource Usage ... 88
3.6.3.4 Resource Usage Search ... 89
3.7 Services ... 90
3.7.1 Available Services ... 90
3.7.2 Searching For Orders ... 91
3.7.2.1 Using The Order Searches ... 91
3.7.2.2 My Orders ... 93
3.7.2.3 My Projects' Orders ... 95
3.8 Toolbar Button Standard Definitions ... 96
3.9 Conditions ... 98
3.10 Searching For Information ... 108
3.10.1 Search ... 108
3.10.1.1 Single Item Report ... 109
3.10.1.2 List Report ... 110
3.10.1.3 Summary Report ... 112
3.10.1.4 Group Report ... 115
3.10.1.5 Editing Search Information ... 117
3.10.1.6 Saving And Reusing Searches ... 119
3.10.2 Report Manager ... 121
3.10.3 History ... 126
3.10.4 Data Explorer ... 127
3.11 Frequently Asked Questions ... 133
3.12 Web Browser Compatability ... 133

Part IV Calpendo Administration Guide 136

4.1 Booking Administration ... 136
4.1.1 Booking Properties ... 137
4.1.2 The Booking Approval Process ... 138
4.1.3 The Booking Requests Page ... 139
4.2 Project Administration ... 143
4.2.1 Project Properties ... 144
4.2.2 The Project Approval Process ... 145
4.2.3 The Project Requests Page ... 146
4.3 User Administration ... 150
4.3.1 User Properties ... 150
4.3.2 The User Approval Process ... 153
4.3.3 The User Requests Page ... 153
4.3.4 User Search ... 158
4.3.5 Creating Users ... 162
4.3.6 Modifying Users ... 163
4.3.7 Changing A User's Settings ... 167
4.3.8 Special User ... 167
4.4 Bookmark Manager ... 168
4.5 Using Calpendo To Send Emails ... 172
4.6 Report Manager ... 176
4.7 Import ... 177
4.7.1 Example Import Files ... 177
4.7.2 The Import Page ... 184
4.8 System Events ... 191

© 2010-2015 Exprodo Software

Part V Calpendo Quick Start Configuration Guide 194

Part VI Calpendo Configuration Guide 198

6.1 Configuration Questions ... 198
6.2 Initial Configuration ... 201
6.3 More Calpendo Configuration ... 203
6.4 Converting From An Existing Booking System ... 205
6.5 Project Configuration ... 205
6.5.1 Project Template ... 206
6.5.2 Configuring Project Properties ... 206
6.5.3 Configuring The Project Approval Process ... 208
6.6 Controlling Bookings ... 210
6.6.1 Choosing Between Time Templates, Booking Rules and Permissions ... 212
6.6.2 Configuring Time Templates ... 215
6.6.2.1 How Time Templates Work ... 215
6.6.2.2 The Time Template Editor ... 217
6.6.2.3 The Time Templates Calendar ... 220
6.6.3 Configuring Booking Rules ... 223
6.6.3.1 How Booking Rules Work ... 223
6.6.3.2 The Booking Rules Editor ... 224
6.6.3.3 Choosing Which Bookings A Rule Applies To ... 226
6.6.3.4 Types of Booking Rule ... 229
6.6.3.4.1 Simple Booking Rule ... 230
6.6.3.4.2 Double Booking Rule ... 236
6.6.3.4.3 Booking Duration Rule ... 243
6.6.3.4.4 Holiday Booking Rule ... 244
6.6.3.4.5 Booking Interval Rule ... 245
6.6.3.4.6 Number Of Bookings Rule ... 246
6.6.3.4.7 Predefined Slots Booking Rule ... 249
6.6.3.4.8 Search Results Booking Rule ... 250
6.6.3.4.9 Total Time Booked Rule ... 251
6.6.3.4.10 Advanced Booking Rule ... 256
6.6.3.5 The Booking Rule Validator ... 264
6.7 Types And Groups ... 265
6.8 Configuring Types And Groups ... 266
6.9 Configuring Resources And Locations ... 270
6.10 Configuring Services and Service Providers ... 276
6.11 Permissions ... 278
6.11.1 How Permissions Work ... 278
6.11.2 The Permissions Editor ... 285
6.11.3 Example Permissions ... 289
6.12 Automatic Emails ... 297
6.12.1 The Automatic Email Editor ... 298
6.12.2 Example Automatic Emails ... 306
6.13 Menu Editor ... 311
6.14 User Authentication Methods ... 322
6.14.1 Authentication Methods ... 322
6.14.2 Authentication Methods Editor ... 324
6.15 FAQ Editor ... 325
6.16 Global Preferences ... 328

© 2010-2015 Exprodo Software

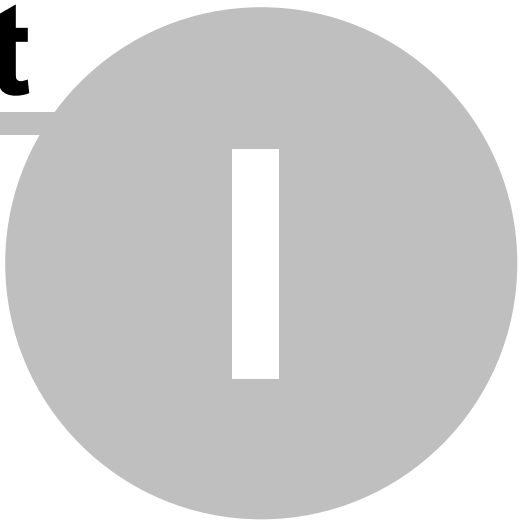
6.16.1 Appearance ... 329
6.16.2 Booking Reminders ... 331
6.16.3 Bookings ... 331
6.16.4 Date And Time ... 337
6.16.5 Email ... 338
6.16.6 General ... 339
6.16.7 Licence ... 340
6.16.8 Menus ... 341
6.16.9 Network Metrics ... 342
6.16.10 Projects ... 343
6.16.11 Resource Usage ... 344
6.16.12 Rules ... 345
6.16.13 Security ... 346
6.16.14 System Usage Statistics ... 348
6.16.15 Time Templates ... 349
6.16.16 Users ... 350
6.17 Bakery ... 351
6.17.1 Property Storage Mechanisms ... 351
6.17.2 Biskit Definitions ... 353
6.17.3 Property Definitions ... 358
6.17.3.1 Biskit Property Definitions ... 363
6.17.3.2 Boolean Property Definitions ... 364
6.17.3.3 Double Property Definitions ... 365
6.17.3.4 Int Property Definitions ... 366
6.17.3.4.1 Mapped Integers ... 366
6.17.3.4.1.1 Bit Sets ... 368
6.17.3.4.1.2 User Roles ... 368
6.17.3.5 Java Enum Property Definitions ... 369
6.17.3.5.1 Java Enum Definitions ... 369
6.17.3.6 Set Property Definitions ... 370
6.17.3.7 String Property Definitions ... 372
6.17.3.7.1 Mapped Strings ... 373
6.17.3.8 String Enum Property Definitions ... 374
6.17.3.8.1 String Enumerations ... 375
6.17.4 Formulae ... 375
6.17.5 Bakery Editior ... 378
6.17.5.1 Bakery Editor Examples ... 384
6.17.5.1.1 Adding Properties ... 384
6.17.5.1.2 Adding Formulaic Properties ... 387
6.17.5.1.3 Adding Properties For File Attachments ... 390
6.17.5.1.4 Adding Properties For Created/Updated/Vesion ... 392
6.17.5.1.5 Adding A New Yes-No Mapped Int Property ... 394
6.17.5.1.6 Creating A New Basic BiskitDef ... 397
6.17.5.1.7 Creating A Set Of BiskitDef (Attachments) ... 398
6.17.5.1.8 Creating A Hierarchy Of BiskitDef ... 402
6.17.5.1.9 Creating A Master-Slave Biskit Relationship ... 407
6.17.5.1.10 Creating An Inheriting BiskitDef (Booking) ... 412
6.18 Layout Editor ... 413
6.19 Backing Up The Database ... 430
6.20 Creating A Read Only Copy ... 432
Part VII Calpendo Installation Guide 436
Part VIII Glossary 446

© 2010-2015 Exprodo Software

Index 451

© 2010-2015 Exprodo Software

Part



1 Getting Started

Calpendo is an intelligent booking system that applies *Booking Rules*[446] to control who can book what and when. It manages information on *projects*[448] associated with each *booking*[446] and automatically sends emails when required.

This documentation was written for version 7.0.52 of **Calpendo** and was last changed on October 29, 2015.

© 2010-2015 Exprodo Software

Part II

2 Version History

This section shows the history of changes over the last few versions of **Calpendo** .

2.1 Changes in Version 7.0

To view changes to other releases go to our release notes at the web site at http://www.exprodo.com/category/release-notes/

2.1.1 Changes For Regular Users

New Functionality

Booking Sub Types

The administrator now has much more control on a *resource* [449] by *resource* basis of the *properties* [448] required to make a *booking* [446]. Therefore the information required on the *booking* pop up may change depending on which *resource* is being booked.

Referenced By Condition

This is an extension of *conditions* [447] that is turned on when the **Show advanced** check box is selected. It allows the addition of a *condition* on the number of times the *biskit* [446] in question is referenced by other *biskits*. E.G. finding all the *projects* [448] that have not had a *booking* in the last 6 months, or that have had more than 10 *bookings* in the last week.

Soft Locking

When a user begins editing anything, if it appears that anybody else is already editing it, then the user will be warned.

Authentication

There's now support for single-sign-on and other authentication methods. This means that users don't need to have separate passwords for **Calpendo** and other systems.

This includes an updated login page [28] and new user registration page [28]. Is also easier to disable user self-registration [322] for those sites that prefer to do this.

Third Party Software

The **About** page now has a link to a page that shows what third party software has been used in making **Calpendo**.

Automatic Timestamps

As well as storing all activity in the audit log, *Bookings* have always directly stored the date and time when they were created and last changed. This has now been extended to other data types including *Projects* and can be added to custom types too [392].

© 2010-2015 Exprodo Software

Automatic Version Tracking

Some built-in data types and custom types can now keep track of their version number. For example, every time a *project* is edited, its version number is increased by one. This can now prevent somebody overwriting other people's changes by making sure when something is saved, the latest version was being edited to begin with.

Changes in Functionality

Exporting Native Excel Files

Native Excel files can now be exported (xls and xlsx). This is particularly useful for countries that use a comma as a decimal separator such as most non-English European countries.

Calendar Pop Up Cancel Button

The *booking* pop-up now shows an additional cancel button to remove confusion between cancelling the booking and cancelling the pop-up.

Faster Start-Up Time

Particularly on slower network connections, there should be a noticeable faster start up time.

2.1.2 Changes For Administrators

New Functionality

Search Results Booking Rule

This new Search Results Booking Rule[250] allows the administrator to check for information on *Biskit Types*[446] other than the **Booking** *Biskit Type*, to help decide whether a *booking*[446] should be allowed and can be configured to fail depending on the number of search results found.. For example, you can create a **Training** *biskit*[446] that captures information about when each person goes on training courses that are required for the facility. Then, create a **Search Results Booking Rule** that checks that the person making the *booking*[446] has received training on the relevant *resource*[449] in the last 6 or 12 months etc.

Booking Sub Types

If custom *properties* are added to *bookings*, different *properties* can now appear for different *resources*. Therefore asking for patient details when booking an MR scanner, but not when booking a conference room. See setting up Resources[270] and Inheritance[357] for more information. If an administrator wishes to move to **Booking Sub Types** for existing *resources* that are currently not using them, then they would be advised to talk to the **Calpendo Support Team** in order to help with the transition including moving across historical data.

Attachments

Files can now be uploaded and stored in the database[390]. For example, documents can be attached to *projects*, such as ethics and other approvals.

© 2010-2015 Exprodo Software

Soft Locking

When a user begins editing anything, if it appears that anybody else is already editing it, then the user will be warned.

Version Control

Version Property *meta-property* [448] used to store an automated integer *property* that stores the version number of each *Biskit*. Then, when you update a *Biskit* with a version, the server compares the given version with the old version.

Automatic Timestamps

As well as storing all activity in the audit log, *Bookings* have always directly stored the date and time when they were created and last changed. This has now been extended to other data types including *Projects* and can be added to custom types too [392].

Authentication

There's now support for single-sign-on and other authentication methods. This means that users don't need to have separate passwords for **Calpendo** and other systems.

This includes an updated login page [28] and new user registration page [28]. Is also easier to disable user self-registration [322] for those sites that prefer to do this.

Changes in Functionality

New Banned Keywords

Added "create" and "update" to list of banned keywords for *property* names

Permissions

Some changes have been made to make *permissions* [448] a little easier to understand.

There used to be a *permission property* called **positive** that was a boolean *property* with values true or false. This has now been renamed [281] to **authorisation** and the options are **Grant permission** and **Deny permission**. It should now be clearer whether a permission that matches will actively allow an action (by granting permission) or stop an action (by denying permission).

Also, support for what we called "definitive permissions" had been dropped. They were a source of confusion. Also, they were not even necessary because every definitive permission could always be represented as two non-definitive permissions.

The **Permissions** [285] page shows the permissions differently now. This is to reflect the fact that permissions for *Booking* will now also apply to subtypes of *Booking*.

© 2010-2015 Exprodo Software

Import

The Import page[184] can now handle importing non-ASCII characters properly, such as accented characters like é, ä and ç etc.

Whenever anything fails to import for some reason, there is now an option of allowing all the items without errors to be imported; previously it was only an all-or-nothing.

Also, the import supports uploading multiple files at once when importing something that includes file attachments.

API Changes

Starting with this release, the **API** (Application Programmer Interface) for connecting other programs to **Calpendo** will be documented. New in this release is support for making complex search queries on the database, and getting back only the columns you need.

There are also some breaking changes to the **API**, so if you were already using it, please contact us.

Old Style Searches

Support for old-style searches has been removed. This should not affect any existing customer because it appears nobody is using the old-style searches any more. If you manage your own **Calpendo** server, then please make sure you keep the database backup you take before the upgrade to 6.3 for a while, just in case we find you need an old search.

New Booking Rule Validation Tests

When saving a *Booking Rule*, there are some new validation tests that are done that look for some misconfiguration's. In particular, if a status is selected under *DoesNotApplyTo*[447] that is not selected in *AppliesTo*[446], then it will give a warning.

Changes To Licence Tab

The Licence[340] tab of the Global Preferences[328] page has changed a little

Time Template Transitions

When creating[57] a *Time Template*[450], a *Time Template* can be specified that applies far into the future, and also in the *near term*[448]. For example, booking may not be allowed more than 6 weeks into the future. It used to be the case that a *Time Template* from 9am to 5pm would only use a rolling transition, which means that if it is now 3pm, then 6 weeks from now the user can book from 9am to 3pm, but not yet 3pm to 5pm.

There's now an option for an instant transition as well as rolling transition. With instant transition, at 9am, the whole of the *Time Template* from 9am to 5pm 6 weeks into the future would become available for booking.

© 2010-2015 Exprodo Software

Add Custom Properties To TemplateGroup

This can be used, together with an **Advanced Booking Rule**, to set up different people as administrators for a *booking* depending on which *Time Template* applied to it.

Boolean Properties

Boolean *properties* have always existed in **Calpendo**, and they allow the specification of a true or false value. However, sometimes, it is necessary to call them something different from "true" and "false", such as "Yes" and "No", or "On" and "Off". The **Bakery** 351 has now been modified to a choice of text to be used to represent "true" and "false" so boolean drop-downs can be customised.

Template Project

There's one special "template" *project* that is used to initialise default values for all *properties* when creating new *projects*. If restrictions on *project* values are setg for example requiring particular *properties* to be set, then those restrictions are no longer applied to the special template *project*.

Custom User Properties

The **New User Registration** page used to ignore some types of *properties*. This has now been extended so it supports all the types that might be required.

Better Booking Rule Error Messages

When saving a *Booking Rule* with an error in one of the *conditions* 447, you are now told which tab the condition is in. So it differentiates between *applies to* 446, *does not apply to* 447, bookings to count etc.

New Ordering Of Booking Rules

Previously a *Booking Rule* had an **Order** *property* which determined which order the *Booking Rules* would run in, now the *Booking Rules* run in the order they are in the **Rule Editor**. *Rules* can be dragged and dropped in the **Rule Editor** to change their order.

Automatic Emails

The **Automatic Emails** 298 page shows the automatic emails differently now. This is to reflect the fact that automatic emails for **Booking** will now also apply to subtypes of **Booking**.

Status Change Buttons On Search Page

When searching for *users*, *projects* or *bookings*, there are now extra buttons that are displayed as a short cut to changing the **status**.

For example, *users* can be easily approved from a regular **User Search** page. This means that the standard **User Approval** page can be replaced with one that only shows a subset of the new *user* requests. This is useful when assigning *user* administration for a group of users to a particular person.

© 2010-2015 Exprodo Software

2.2 Changes In Version 7.0.1 - 7.0.45

To view changes to other releases go to our release notes at the web site at http://www.exprodo.com/category/release-notes/

2.2.1 Changes For Regular Users

New Functionality

Resource Names Tool-tip

When the mouse is run over the a resource heading in a calendar a tool tip appears displaying the full resource name.

New Report Type

Add support for search for single *biskit*.

Resource Order

Add support for reordering resources in the resource selection pop-up using up and down arrows.

User Settings

- Add support for opting out of automatic emails. This does not apply to booking reminders.
- Add support for a user to define their calendar start/finish hour and day of week

New Group Report Aggregation

Add support for a group report aggregation of "value at minimum" and "value at maximum". For example, if a user has a *biskit* that stores a bank account balance, they can add a column to the report for the date, and give it an aggregation of "value at minimum" of "account balance". This means it will then produce the date on which the account balance is at its minimum.

ICal Support

Add support for iCal bookings feed to send bookings to other calendar applications (eg Google Calendar, iCal, Outlook, Android, iOS).

Calendars

Add button to hide/show the nav/filter panel on all calendars.

Services

Added support for **Services**.

© 2010-2015 Exprodo Software

Changes in Functionality

Month View

The **Booking Calendar Month View** shows the *booking* finish time as well as the start time.

Saving Bookings

Booking popup does not allow a *booking* to be saved while the finish time is later than start time.

Downloading Attachments

Prevent attachment download if the user does not have access to its referring *biskit* or *property*

Report Description

A report description has been added to emailed reports.

Checked Box Editing

When editing using a list view, by checking the boxes of the elements in the list view, if the *properties* have a **Layout** or are **Grouped** then they will be displayed in a hierarchical view rather than a long list of *properties*.

2.2.2 Changes For Administrators

New Functionality

Time Template Calendar Month View

The **Time Template Calendar** now has a **Month View**

Automatic Email Attachments

Add the ability to have attachments on A*utomatic Emails* by being able to reference them from the Email body.

© 2010-2015 Exprodo Software

Custom Search Page

- Add support for int/long/double/string filter widgets in custom search page.
- Upgrade custom search pages to support form-based custom search widgets that appear in a panel down the left rather than a single-line at the top.
- Add support for custom search page search term that checks for a range of numeric values.
- Text search boxes in custom search pages can now use "contains" and "matches" as well as "equals" checks
- Custom search boxes can now have user-defined labels

Calpendo Lite

- Apply limits to number of custom *properties* when using licence of type **Lite**.
- Apply limits to **Rules** when licence of type **Lite**.
- Stop **Calpendo Lite** from creating advanced **Rules.**
- **TemplateGroup** should now stop you creating once three have been created.

Conditions

- Add support for regular expression matching in conditions.
- Add support for is-ancester-of, is-not-ancestor-of, is-descendant-of and is-not-descendant-of condition relations that can apply to biskits configured as a hierarchy.
- Add support for non-Booking formulaic properties in rule conditions.

New User Status Lurker

Add new "lurker" user status who can't log in, but can get emails. Other users who can't log in can no longer receive emails

Dynamic Notes and Waterfalls

Add support for dynamic notes and waterfall. This lets a user easily add attach a text note to something, and see a waterfall of such notes. This is set up in the **Layout Editor**.

User Names

Add support for login names with more exotic characters, such as @!:/(). Previously, only alphanumeric names were allowed.

© 2010-2015 Exprodo Software

Resource Manager Group

Added a new *property* to *Resource* called **Resource.managerUserGroup** to allow an administrator to specify a group of *users* that can make *bookings* for any *project* on that resource and can approve/deny/cancel *bookings* on that *resource* once the appropriate *permissions* have been set up. This allows a group of people to very quickly be set up as administrators for a number of *resources*.

Email Address Validation

Add validation of any string property that contains an email address to ensure the email address is legal.

Set Automatic Email Reply Address

Add support for setting automatic email reply-to address in **Global Preferences.**

Recording of Schema Changes

Search for *biskits* of type **DatabaseUpgrade.**
This records both updates done via the bakery in the regular way with **Update DB Schema**, but also any **SQL** executed by using the custom **SQL** tool that you see in the bakery when logging in as root.
This should correctly record queries that work properly and queries that fail, along with the mysql error.

Services

Added support for **Services**.

Storing Images to be Accessed by CSS

Create an instance of **PublicResource**, with an attachment, and make sure user **nobody** has read and exists on both the **PublicResource** and the **Attachment**.
Then, the URL slug "/downloads?type=public&name=FOO" will yield the attachment, where you must replace FOO with the name given to the public resource (which must be unique)
E.g. "https://thing.calpendo.com/downloads/type=public&name=FOO"

Resource Groups

Resources can now be put into *Groups* in the **Resource Editor** in addition to the **Types And Groups Editor**.

Groups

Resources, *Users* and *Projects* can be added and removed from *Groups* en mass using the **Collection Editor**.

© 2010-2015 Exprodo Software

Changes in Functionality

New Banned Keywords

Added "procedure" to the list of banned keywords for *property* names

Import

- Add support for the import of **Null** values.
- Add support for importing a date/time that is a number of seconds since 1970.
- Add support for importing **id** properties.
- Generate an error on importing an unrecognised **BitSet** value.
- Add support for import page to be able to choose which lines are imported.
- *Property* names in the heading are now matched case insensitive.
- Names for a **MappedInt** and **MappedString** are now matched case insensitive.
- On updating data, allow key properties to be modified by adding a second column for the same property, and defining that column to be "normal" rather than a key.
- Allow the *Biskit type* to be narrowed on each import line. Eg, if you import a mixture of different *booking* subtypes or *bookings* with daily repeat and weekly repeats.
- Add support for warnings on import as well as errors.
- Provide link in each column to show the problems from that column.
- Provide link to re-display import problems even after it has been dismissed.
- Improve ability for server to send errors to user during import.
- Boolean *properties* now accept custom text for true/false e.g. yes/no.
- Booking import checks that the biskit type matches what the *resource* wants.
- Make import treat empty email address as meaning null email address.
- Set column to not be ignored when column path manually set.
- Click in an Import error pop-up now auto-scrolls preview to that item
- Import now gives an error if you have too many columns for the same path
- Give better error message when you try to import a binary file instead of CSV
- Add support for import to hide some rows/columns from import preview table
- Import now shows spreadsheet-style column names as well as column numbers
- Imported files now accept column prefixes of "**KEY**:" and "**IGNORE**:".
- The import now reports the number of lines imported.
- Improved error message appears when a user logs out and back in, and then tries to import a file they had uploaded before they logged out.

© 2010-2015 Exprodo Software

Optimisation

Search would sometimes do more work than necessary.

Bakery

- Prevent a user from setting parent of a *BiskitDef* to itself.
- **ResourceType** and **ProjectType** may now have custom properties.
  - Custom formulaic properties are now syntax checked for balanced () brackets
  - Set the name of the default id column in bakery to include the biskitType
  - Allow formulae to reference primary keys using pseudo id, eg as [foo.id]
  - Add support for automatic recording on a biskit of who created it and was the last to update it. Create a user biskit property on the target biskit type, mark it as automated, and set as the creator/updator property.
  - Add support for string properties that contain ab RGB colour
  - Add new UserGroup.self property

Authorisation Methods

Make external authorisation methods show before non-external on the login page.

Database Dump

Database dump file names include date, time, version,system name and server hostname (as specified in the global preferences emailed base URL)

Layout Editor

Add Layout Editor to all admin menus that don't already have it.

Total Time Booked Rule

- Rule message to include time allowed as well as time used.
- Add support for negative quota to mean no limit in Total Time Booked Rule

Number of Bookings Rule

Add support for negative quota to mean no limit in Number of Bookings Rule

Global Preferences

Removed the global preferences property "send old search usage home". Support for old-style searches was removed quite some time ago, so this preferences setting no longer serves any purpose.

© 2010-2015 Exprodo Software

2.3 Changes in Version 7.0.46-7.0.55

To view changes to other releases go to our release notes at the web site at http://www.exprodo.com/category/release-notes/

2.3.1 Changes For Regular Users

New Functionality

None Currently

Changes in Functionality

Update Displayed Breadcrumb On Any Page Change Event

- The breadcrumb is the text in the top-right corner of the page that shows which page the user is on
- Previously, it was only on selecting a new menu item that the breadcrumb changed.
- The biggest impact of this is that when clicking from one biskit to another, the breadcrumb can now change.

Emailed Reports Have Attachment

Make reports that are emailed, sent as an attachment by default (was sent inline).

Emailed Opt Out

Opting out of automatic emails now means the user opts out of reminder emails too

2.3.2 Changes For Administrators

New Functionality

Charts

If an administrator want to know about charts, please ask. They are currently hidden from general use, but can easily be enabled.

© 2010-2015 Exprodo Software

Rich Text Editor

Linking to a menu item.

- Highlight some text, click the new "Menu" drop-down button on the rich text toolbar and select a menu item.
- The highlighted text will then become a hyperlink that will run the same thing as the selected menu item.
- NOTE: You must choose a menu item that will be on the menu of the person who clicks on the link.

Add links to attachments & public resources.

- Allows the embedding of hyperlinks to files within any String property that is marked in the bakery as containing HTML.

History Page, Create Copy Of Old Versions Of A Biskit

- Click on a particular version in the history of a biskit, and there is now a "Create Copy" button. This offers the chance to create a new biskit using the old values.
- Note that some properties (those that are a set or list) are not recorded in the audit log, and so will not be created in the new copy.

Biskit Tree Viewer Page

This provides a way to display a number of HTML pages with a navigation panel for users to choose which page to view.

Initial Page to Load

The **Global Preference** for the initial page to display on loading the application, now has a button and drop-down allowing selection from a list of all known pages, all items on the menu and recently accessed pages by the current user.

User Biskit Changes

The date/time of a users last login and the IP address the user logged in from are now recorded directly on the user allowing easier reporting on when users have logged in.

Advanced Rules

Advanced rules have a new option to change behaviour with repeat bookings . For repeat bookings, rules are normally run once for every instance of the repeat booking. There is now an option on advanced rules to run it only once instead.

© 2010-2015 Exprodo Software

Changes in Functionality

Public Resource

Added a property to PublicResource to show what the public URL is. PublicResource is useful to upload a file too, that anybody should be able to download, even if not logged in.

Dumping Database

No longer be able to dump the database if the administrator does not have "Dump Database" permission plus unfettered READ and EXISTS on all biskit types.

New option for a simple choice of which tables to dump.

Custom Search Page

Any custom search page that the selection from a "bit set" property (such as user roles) now allows the choice of items that have any, all, none or not all of the selected items.

History Page

Make history page default to asking for history for the last 8 hours.

Multiple OneToMany Properties on a Biskit

- Were there is a biskit C ("child") which has multiple biskit properties referencing some other biskit P ("parent"). Make it easy for somebody looking at a P to be able to see which C reference it. So there are properties on C of type manyToOne, and add oneToMany properties on P.
- The problem is that oneToMany properties are normally cascaded for deletion, so editing P such a child C is removed from one of its oneToMany properties would mean the C would be deleted.
- The Bakery now allows the specification of a ReferenceDeletionOption on oneToMany properties. This should be CASCADE for the old behaviour, and SET_NULL to indicate the parent does not own the child.
- Also, the UI has been modified so that there is no mechanism in the UI for adding a new C or removing an old C from a oneToMany property on P if that property has its ReferenceDeletionOption set to SET_NULL.
- The BiskitDef validation done by the bakery now checks for multiple oneToMany properties with the same child type. It then complains if there are many such properties and any of them don't have their ReferenceDeletionOption set to SET_NULL.

© 2010-2015 Exprodo Software

Record Which Pages A User Has Used For Exception Reports

- The last few application pages visited are now remembered by the web browser.
- Whenever the client detects an error, the event that is logged in the database now includes the record of which pages were visited recently.
- If the **Global Preferences** allow for error reports to be emailed back to Exprodo Software, we will now receive this record telling us what was being done when the error occurred.
- This will make it easier to fix bugs by automatically giving us better information about the context of an error.
- The information we receive will be:
  - The contents of your browser's address bar after the "#"
  - The label of the menu item for your current page (the breadcrumb)
  - The name of the part of the program that generated the page
- If anybody has concerns over privacy, please let us know. But you can prevent this information being sent to us by disabling the automatic exception reports in Global Preferences. Also, if your browser does not suffer from an unexpected error, then this new information remains within your browser and is not sent anywhere.

Audit Log

Show full audit log detail when click on entry in history page

Hierarchical Resource Types

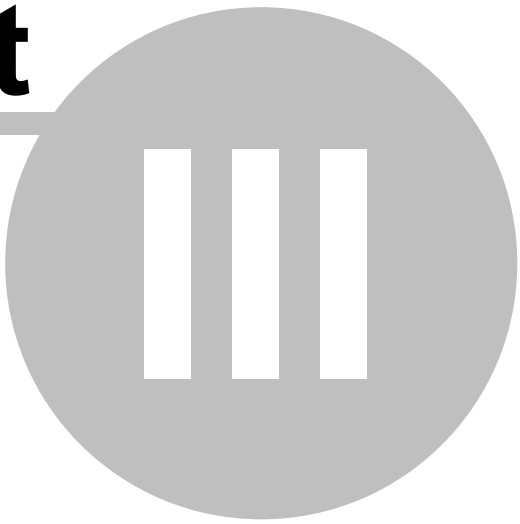
- This can be enabled by adding two properties to ResourceType. One of type Biskit/ many-to-one to store a reference to a "parent" ResourceType, and another of type Set/ one-to-many to store a reference to child resource types.
- This then allows the creation of a hierarchy of resource types so you can classify instruments. For example, creating a resource type of "microscope", with subtypes "Electron Microscope" and "Optical Microscope", with subtypes of Electron Microscope of "SEM" and "TEM". etc.
- This support also allows the Booking popup to view the Resource Hierarchy when choosing a Resource.

New System Events

A system event is now logged whenever an automatic email or reminder is not sent to a user due to them having opted out of automatic emails.

© 2010-2015 Exprodo Software

Part



3 Calpendo User Guide

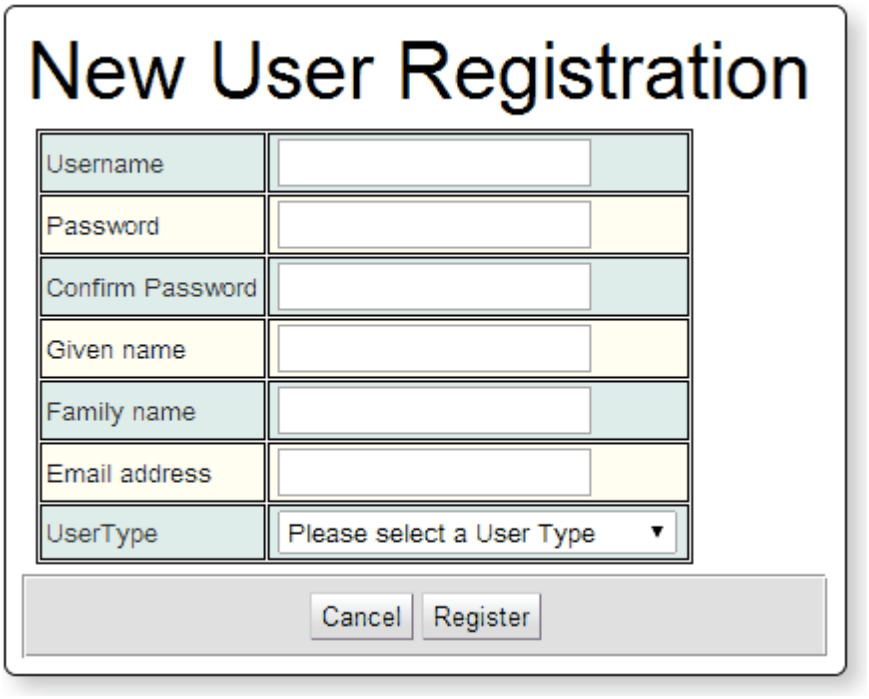
This guide explains how to use **Calpendo** to see or make *bookings* [446], create or edit *projects* [448] and how to search for information such as how much time a user has booked altogether.

3.1 Getting A User Account

Local Calpendo Account

To get a local user account, first open a web browser. **Calpendo** supports the most popular web browsers - see Web Browser Compatibility [133] for the full list. The administrator should have told you the address of the **Calpendo**. When **Calpendo** is started, the opening screen will look something like this:

Press the **Register new user** button to see the **New User Registration** form shown here



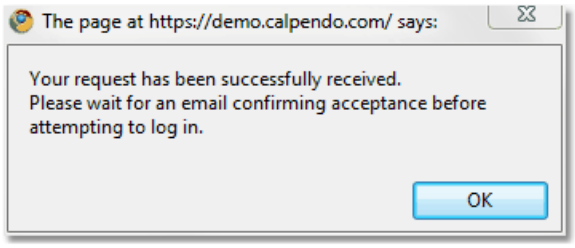
© 2010-2015 Exprodo Software

Enter the desired login name, which can consist of letters, numbers, and other characters such as @!:/(). Enter the password that is required by the user with conformation, then enter the given name (usually the first name) and family name (usually the last name). Finally enter the required email address. It is important that the email address is entered correctly because **Calpendo** will send an email to confirm it has received the registration request and again to inform when or if the request has been approved and the user can log in. If User Types have been configured you will need to choose one.

Depending on how **Calpendo** has been configured, other information may need to be selected.

When all values are entered, press the **Register** button.

If the requested login name, password and email address are acceptable, then a message like this will appear:


The page at https://demo.calpendo.com/ says:
Your request has been successfully received.
Please wait for an email confirming acceptance before attempting to log in.
OK

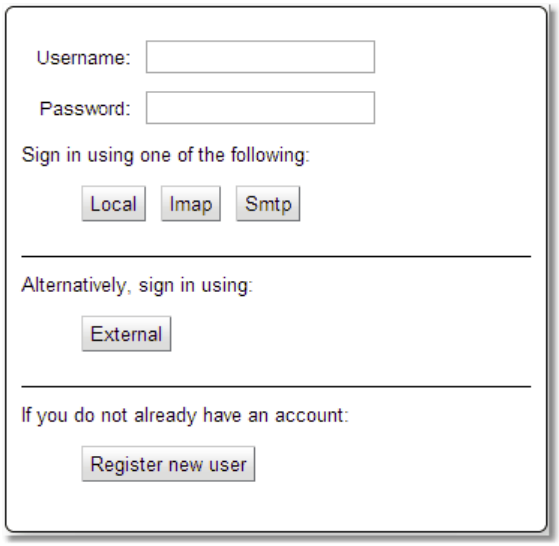
This means that the registration request has been received, but the user cannot log in yet. They will receive an email to tell them when they can log in using the login name and password selected. Note that the administrator will not know any passwords, nor have any way to find out what they are. However, the administrator is able to reset a password if required.

When a user first logs in to **Calpendo**, they will see the Bookings Calendar[36].

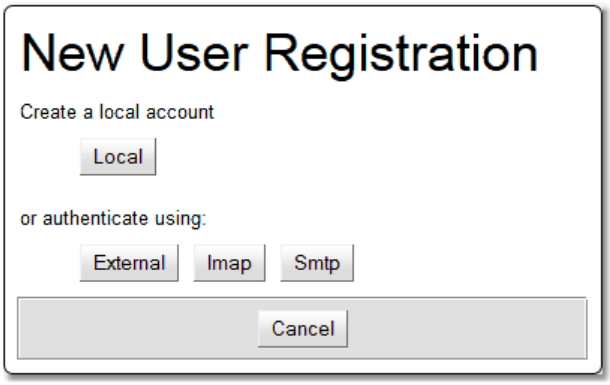
© 2010-2015 Exprodo Software

Single Sign On Calpendo Account

Some customer environments use a single sign on. To access **Calpendo** using a single sign system a user would need to get authenticated by the single sign on system. To log in click on the button for your sign in system or the **Register new user** button to register using a specific authentication method. The names on the button(s) will have been chosen by the administrator when single sign on authentication is set up so may be different.



If registering a new user choose the system to be registered by. If using a non-local method the user will need to be authenticated by that system before filling in the new registration form.



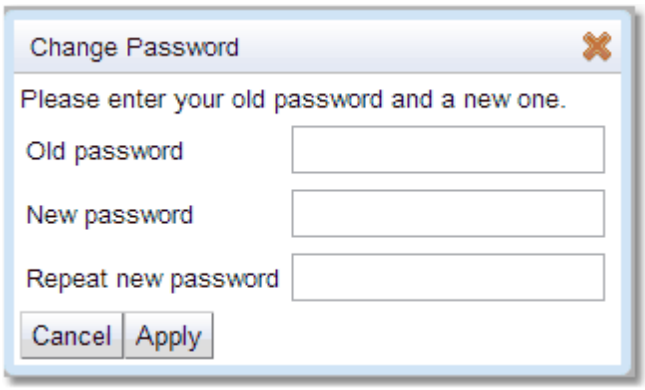
And then complete the **New User Registration** form, to create an account within **Calpendo.**

Once this account has been accepted by the administrator of **Calpendo** then the user will be able to login. This enables administrators to control whom has access to **Calpendo.**

© 2010-2015 Exprodo Software

3.2 Change Password

If logged in using a local authentication method, in the top-right corner of every page of **Calpendo**, there is a **Change Password** link. Click on this if the user would like to change their password. A pop-up will appear that requests the old password and two copies of the new password. Press the **Apply** button to change the password.



© 2010-2015 Exprodo Software

3.3 User Settings

The *User Settings* 450 page configures various options that control how **Calpendo** looks. Access it by going to the **Settings** link that is shown in the top-right corner of every page in **Calpendo**.

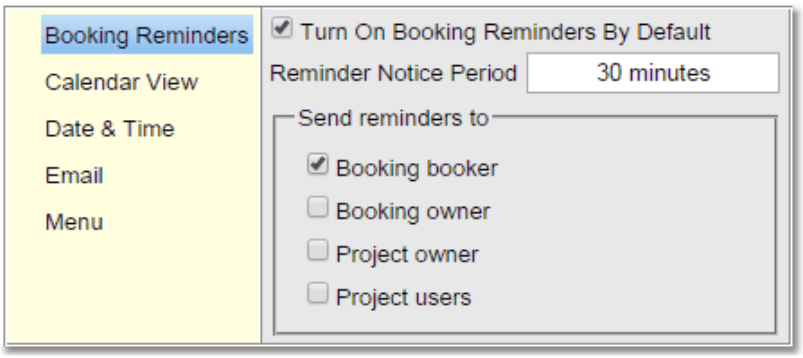
This will show a *users settings* with several tabs, as described below. Once the required settings have been changed then press the **Save** button to save all the changes.

3.3.1 Calpendo User Settings

Booking Reminder

When a *booking* 446 is created, the user can choose whether a reminder email should be sent ahead of the time that has been booked. This is described in Creating Bookings 52. Each of the settings on the reminder email section are given default values, and this **Booking Reminders** tab of the *User Settings* 450 page sets up what those defaults are.

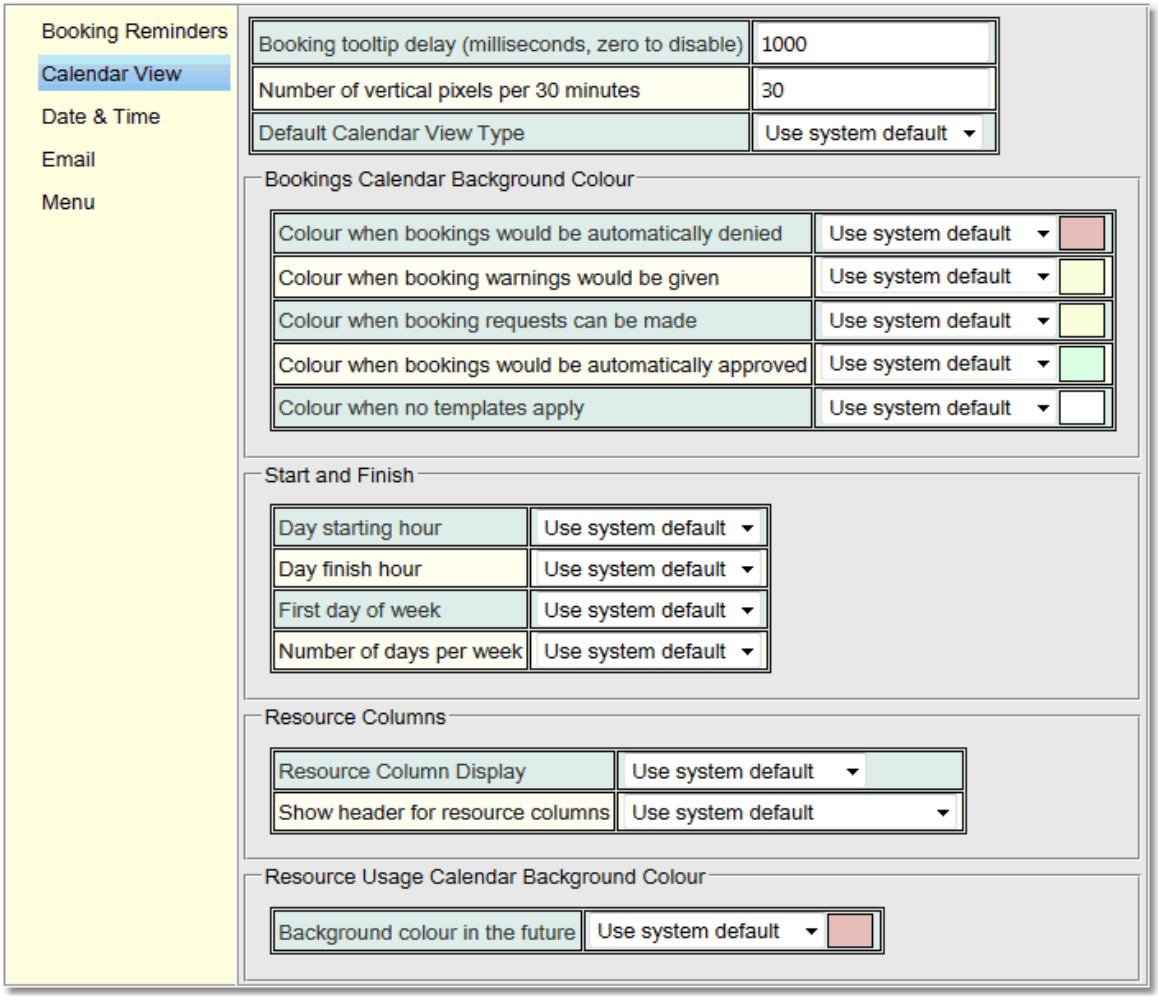
If it is preferred that *booking* reminders are to be off by default when creating a new *booking*, then just untick **Enable Automatic Booking Reminders**. Set up the default for who should receive an email, and the notice period ahead of the *booking* when the email should be sent.



© 2010-2015 Exprodo Software

Calendar View

This tab customises some aspects of the way the *Bookings Calendar*[446] is displayed.



The top section deals with some general viewing options.

When the mouse hovers over a *booking*, a tool tip appears that gives information about the *bookings* content. These tool tips can be disabled or the period of time the mouse must be stationary before the tool tip shows can be set, by changing the **Booking tooltip delay**. Set this to zero to disable the *booking* tool tips, otherwise set this to a number of milliseconds.

Choose the amount of space allocated to each hour of the day in the calendar by changing the **Number of vertical pixels per 30 minutes**. The default is 30. If a larger number is selected, then the *calendar* will take up more vertical space. Or it can be shrunk by reducing the number. Note that this does **not** change the font size used to display on the calendar.

Choose whether to use the system default view for the *calender* when **Calpendo** first starts or whether it is to start in Day, Week or Month view.

© 2010-2015 Exprodo Software

The **Bookings Calendar Background Colour** section allows specification of the colours used to display *Time Templates*[449] in the background, if requested on the *Bookings Calendar*. Choose the colours depending on whether *bookings* would be *automatically denied*[446], *automatically approved*[446], given a warning, *requested*[449] or when no *Time Template* applies.

The Start and Finish section allow the user to define which days of the week to be displayed as well as the hours on their calendar. So for example, the *calender* will only show from 8am to 6pm and from Monday to Friday if the building was always closed outside those hours.

The **Resource Columns** section:
Lets you select how to allocate columns for each *resource*[449].

a. **Share When Possible** means bookings for different *resources* will be put into the same column as long as they do not occur at the same time.

b. **Separate** means *bookings* for different *resources* always appear in separate columns. If there are no *bookings* for a particular *resource*, then no column will be shown for that *resource*.

c. **Always Present** means *bookings* for different *resources* always appear in separate columns, and the column will display even if there happens to be no *bookings*

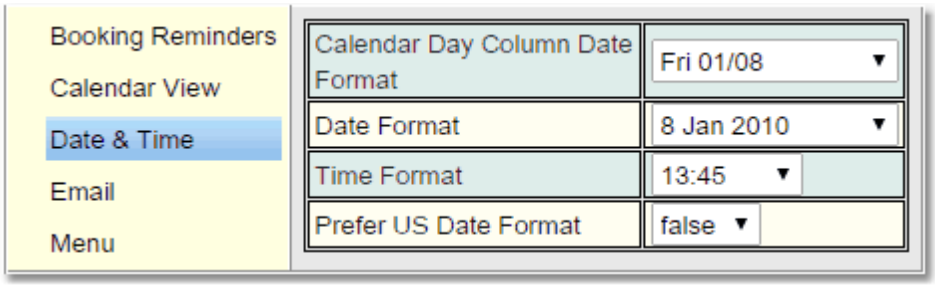
This also allows the definition of whether the *Bookings* and *Time Templates Calendars*[450] will have a heading at the top of each column for each *resource* displayed on the *calendar*.

Finally set the colour to appear on the *Resource Usage Calendar*[449] for the future when creating *resource usage*[449] reports would be denied.

© 2010-2015 Exprodo Software

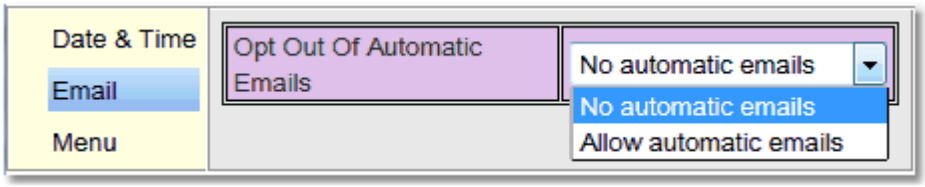
3.3.2 Date & Time

This tab allows selection of the format to use for displaying the day and date at the top of each day in the *Bookings Calendar*[446], and also the date and time format used everywhere else. Also specify whether US date format is preferred.



3.3.3 Email

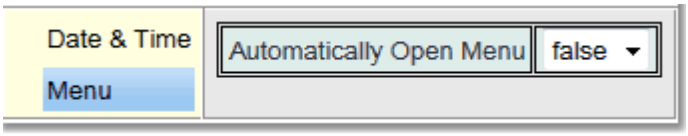
This tab allows the user to choose whether to opt out of automatic emails (not reminders).



3.3.4 Menu

This tab allows the user to choose how the menu operates. Either configure it so that each sub menu opens up just by placing the mouse over a menu, or that the user must click on a menu for its sub menu to open.

Also, **Calpendo** can show different menus to different user roles. If the user has sufficient *permission*[448] to change their menu, then a drop-down that lets you select which menu you should see will be shown.



© 2010-2015 Exprodo Software

3.4 Bookings

A *booking*[446] in **Calpendo** is for a period of time and always for one *resource*[449]. A *resource* is a room, piece of equipment or person that is bookable. If two *resources* need to be booked simultaneously, then that requires two **Calpendo** *bookings*. *Bookings* may represent an exclusive use of a resource[236], or there might be non-exclusive use of some *resources*, depending on how **Calpendo** has been configured.

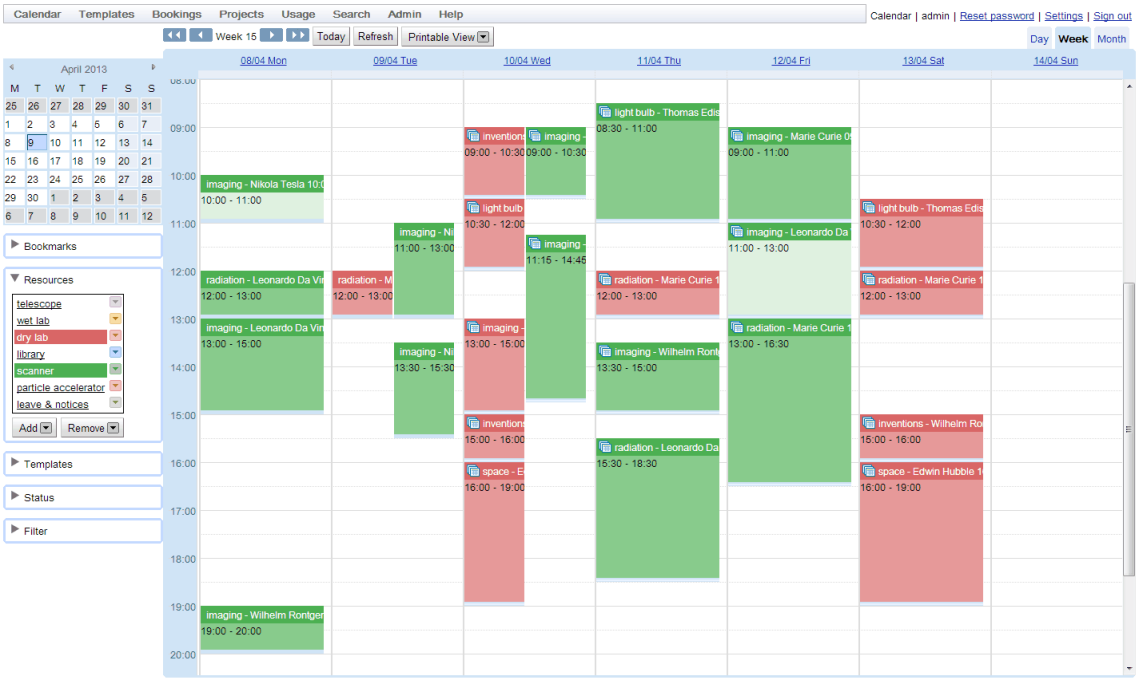
Some *resources* may require that there is a project[69] associated with each *booking*.

Bookings may be repeating (for example, every day or every week) and it's possible that a user may be able to create *repeat*[449] *bookings* only on some *resources* or not at all, depending on the Booking Rules[223] and Permissions[278] that have been set up.

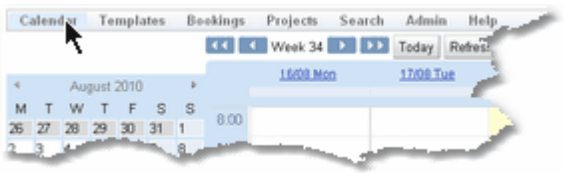
Bookings may have to use predefined slots where both the time and the length of the *booking* has been defined by the administrator.

3.4.1 Bookings Calendar

When **Calpendo** is first started, the user will see the *Bookings Calendar*[446] showing a week of *bookings*[446].



By default, the *Bookings Calendar* appears on the **Calpendo** menu here:

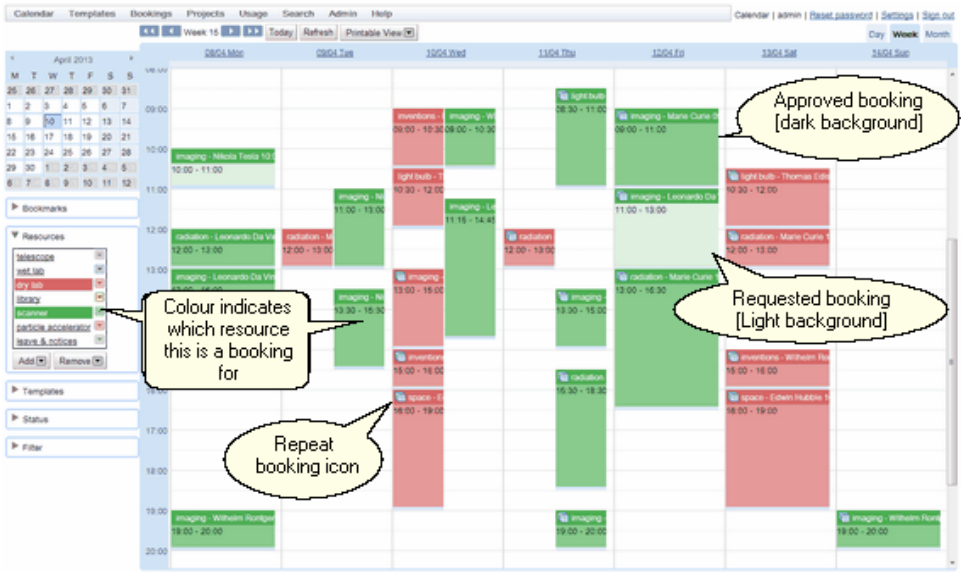


However, the administrator may have configured **Calpendo** so that the menu is different.

© 2010-2015 Exprodo Software

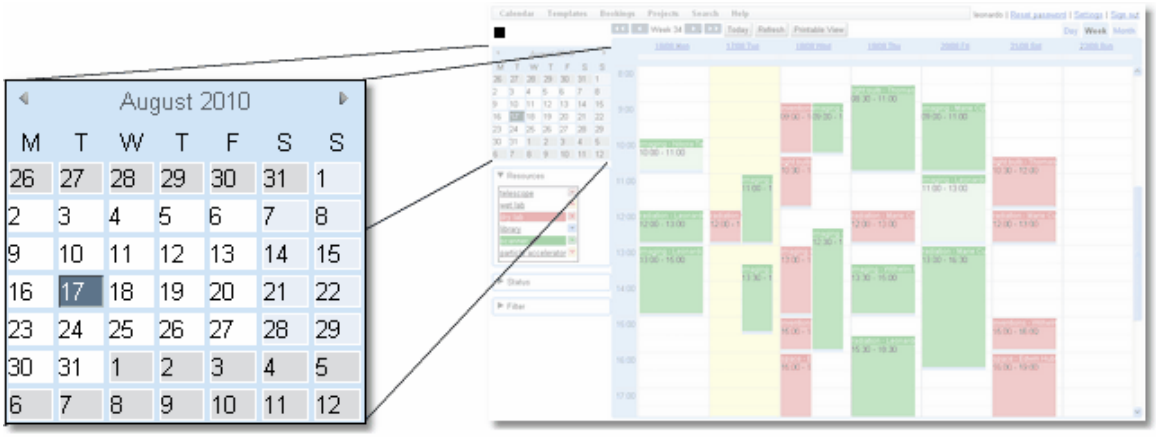
3.4.1.1 Exploring The Bookings Calendar

The colours on the *Bookings Calendar*[446] are used to indicate the *resource*[449] being booked, and the shade tells you whether a *booking*[446] is a *request*[449] or if it has been *approved*[446]. A dark *booking* is *approved*, while a light *booking* is usually a *request* (although if **cancelled** or **denied** *bookings* are requested to be displayed, they also display in a light shade). This screen shot shows a week's *bookings* for two *resources*, the *bookings* for one *resource* in green and the other in red.


Approved booking [dark background]
Requested booking [Light background]
Colour indicates which resource this is a booking for
Repeat booking icon

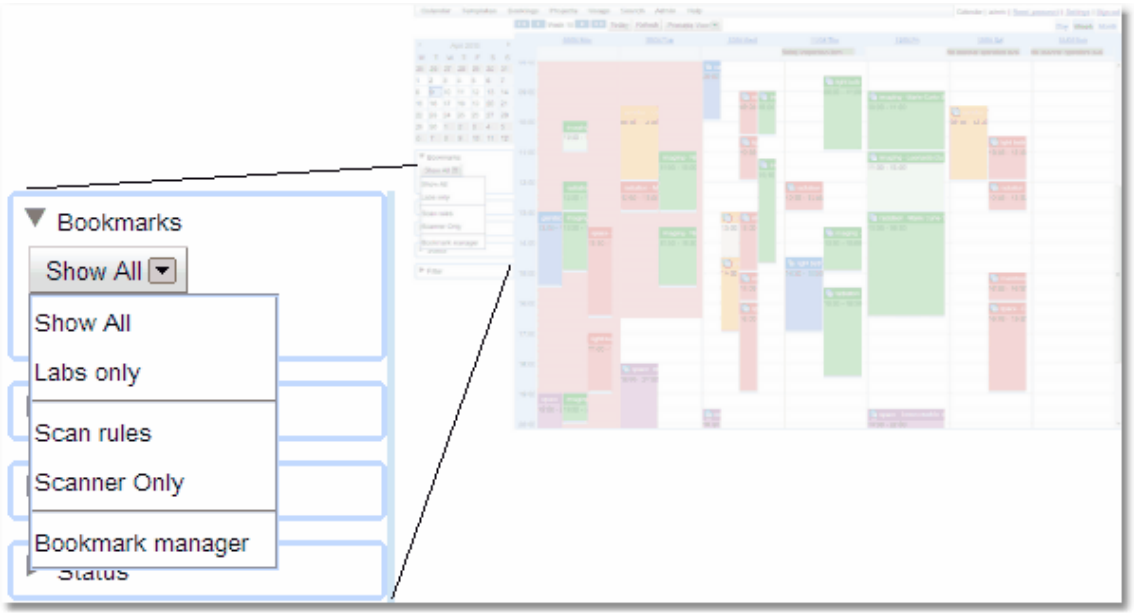
The navigator and filters/selectors can be removed/returned by pressing the arrow button found in the top left hand corner of the screen. This can be found on all calendars.

The month navigator quickly changes the date being displayed. Click on any date to display *bookings* for that date. This will work the same regardless of whether the view is for a day, week or month of *bookings*. Today's date is shown with a dark background.

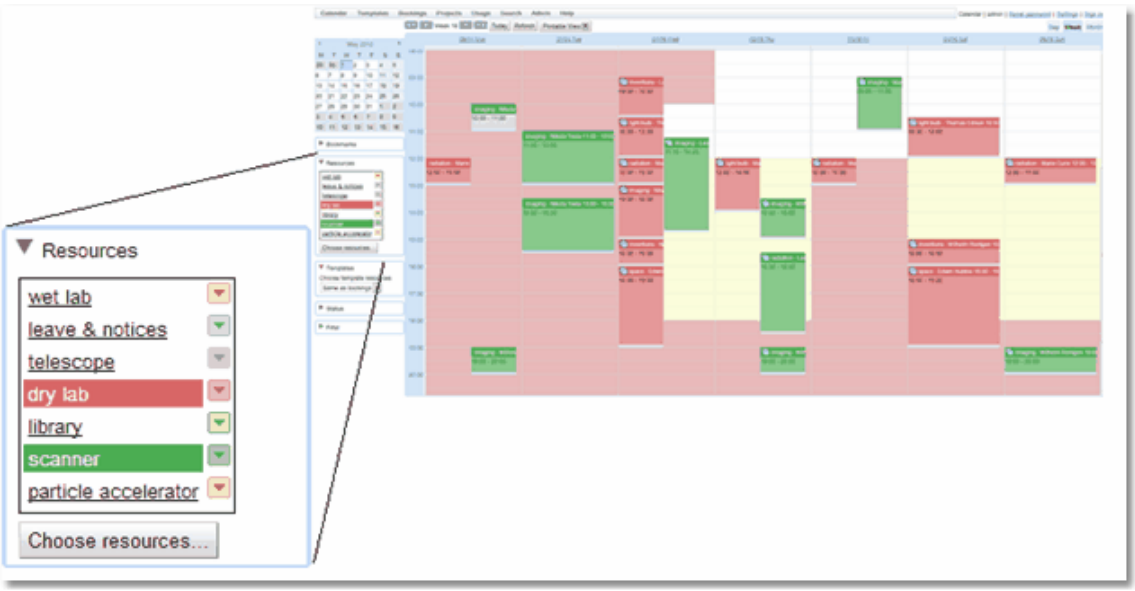


© 2010-2015 Exprodo Software

The *Bookmark*[447] selector allows users to display only the *resources* they wish to see. *Bookmarks* can be system wide or on a per user basis. Click on any *bookmark* to see the *resources* that *bookmark* shows.For more information on how to set up and use *bookmarks* see the Bookmark Manager[168] chapter.

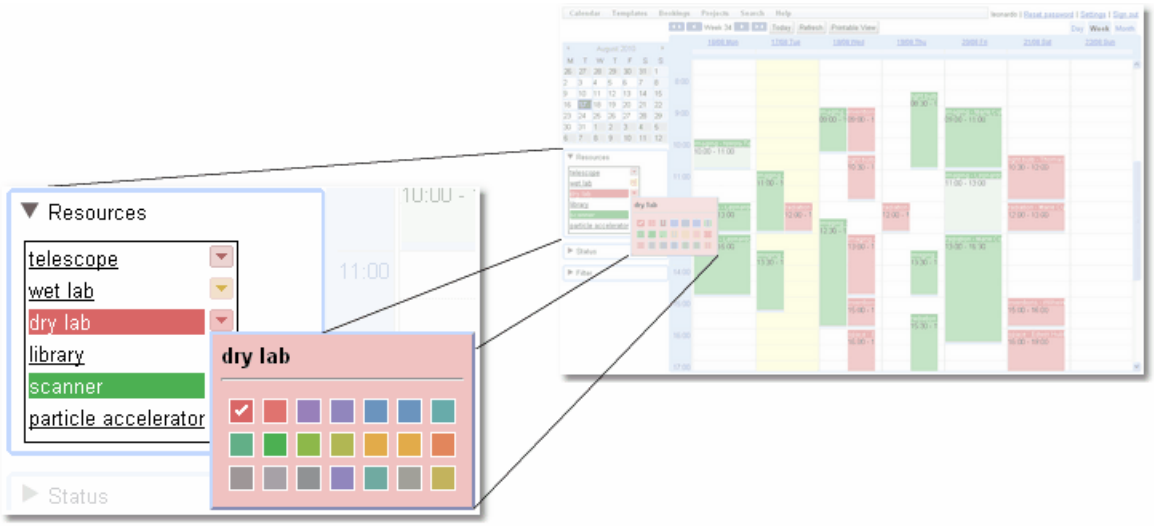


The **Resource Selector** shows the available *resources*, which ones are currently selected, and what their colours are. Just click on the name of a *resource* to hide or display the *bookings* for that *resource*. If the background colour shows then that *resource* will be on the *calendar*. If all the *resources* are not required in view click the **Choose resources...** button and use the **Resource Selector** to pick the *resources* to be shown on the list. For a description of how the **Resource Selector** works read the Resource Selection[44] section at the end of this chapter.

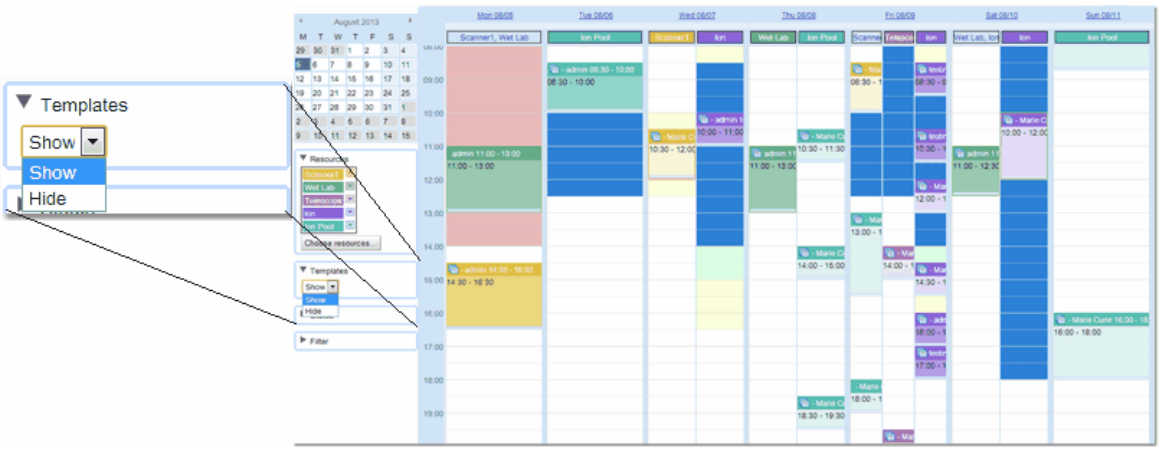


© 2010-2015 Exprodo Software

To change the colour used for any *resource*, click the button to the right of the *resource* name. This shows a pop-up where you can select from the available colours.

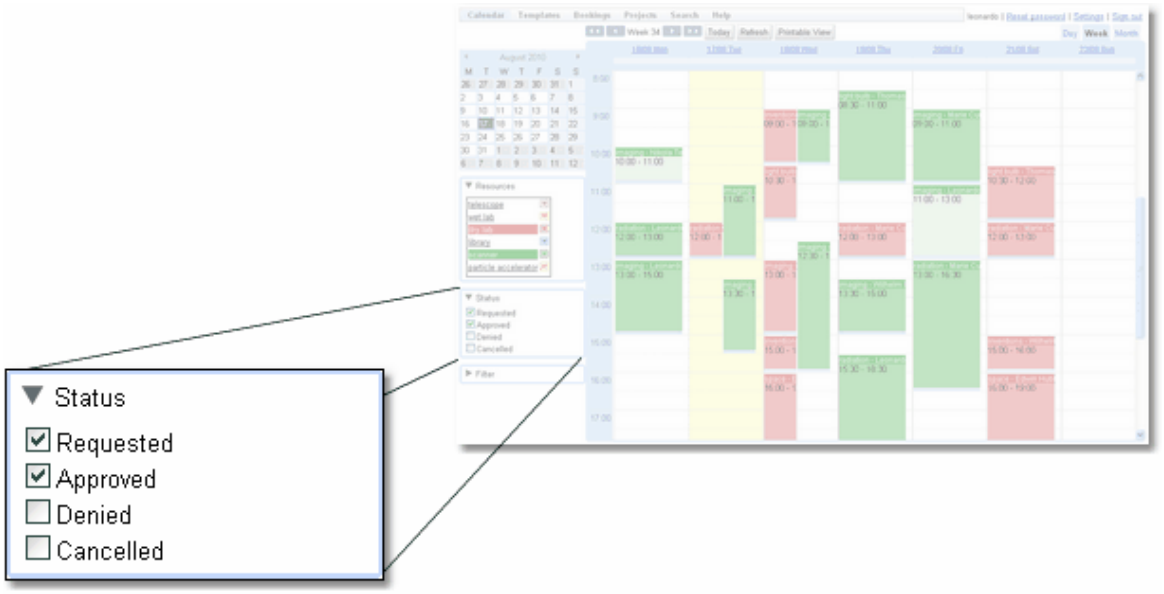


The *Time Templates* option allows you to decide whether to show or hide basic *Time Template* information for all *resources*. This allows the user to quickly see for a *resource* which *Time Templates* are being used. Areas in red are either in the past or *bookings* are not allowed, in yellow, *bookings* may be requested awaiting acceptance although a warning may be issued, in green when *bookings* will be automatically accepted or white where no *Time Template* is present. (The colours are the defaults and may be changed in User Settings 32. In the example below the *Requested* colour has been changed to blue to make those areas stand out from the warning areas.). If the *calendar* shows *Time Templates* for multiple *resources* in the same column, then it will show the background colour to represent the most lenient restriction for all the *resources* it's showing *Time Templates* for (no *Time Template* being the least restrictive). Which means if the background is red, then none of the *resources* displayed in that column can not be booked. If it's white, yellow (or blue in the example below) or green, then the user will be able to book for at least one of the *resources* displayed in that column. Moving your cursor over an area where *Time Templates* are active, for the current user, will provide a tool tip with any message(s) the administrator has defined for the *Time Template*(s).

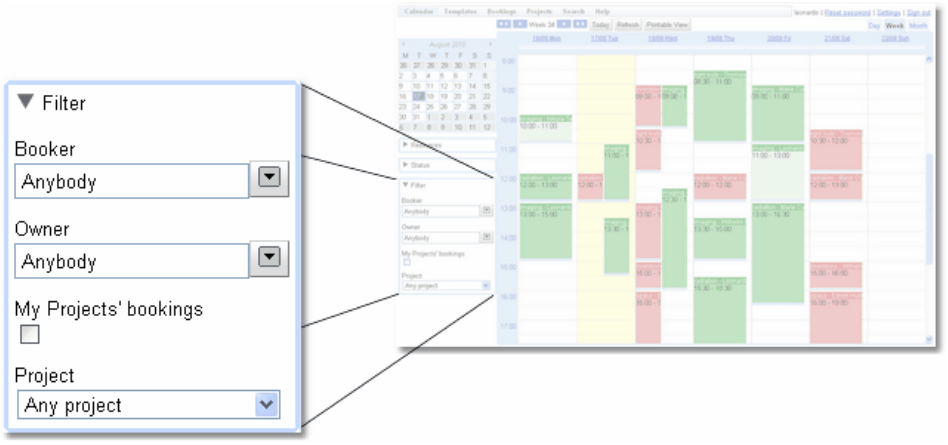


© 2010-2015 Exprodo Software

Normally, the *bookings calendar* displays *bookings* that are *approved* or *requested*, but ignores **denied** or **cancelled** bookings. Click on the **Status** label, to open up the **Status Selector**. Here choose the statuses to be displayed.



Click on the **Filter** label to provide more detailed filtering options. Using a filter can restrict which *bookings* are displayed by selecting who booked or who owns each *booking*.

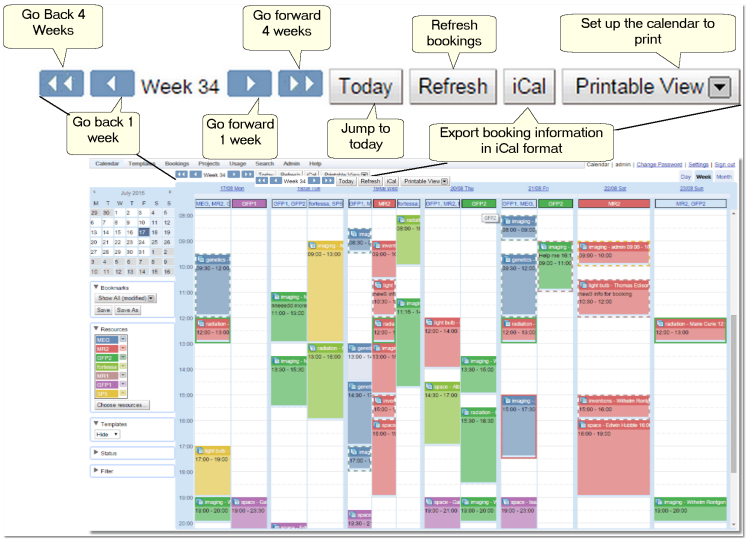


Click **My Project's Bookings** to show only *bookings* for *projects* that the user is associated with.

Select a particular *project* to show only *bookings* for that *project*.

© 2010-2015 Exprodo Software

The button bar at the top of the *calendar* allows movement through time, refreshes the currently displayed *bookings,* exports **iCal** information or switches to a view suitable for sending to a printer.



The forwards and backwards buttons change how far forwards and backwards they go depending on which view is currently shown, day, week or month view.

In a day view, the buttons go forwards or backwards one day or one week.

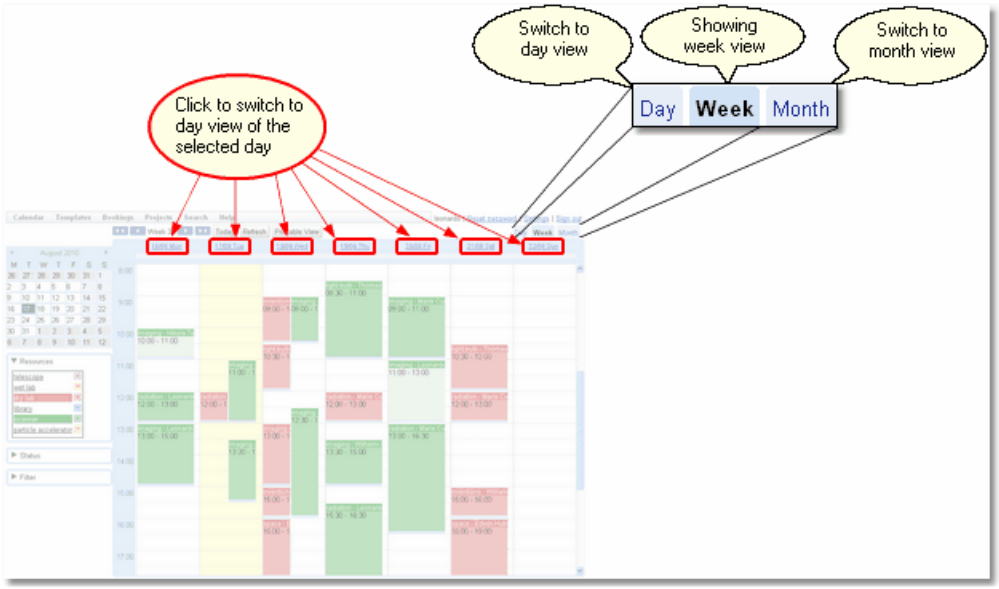
In a week view, they go forwards and backwards one week or one month.

In a month view, they go forwards and backwards one month or one year.

The **Today** button will always jump to today.

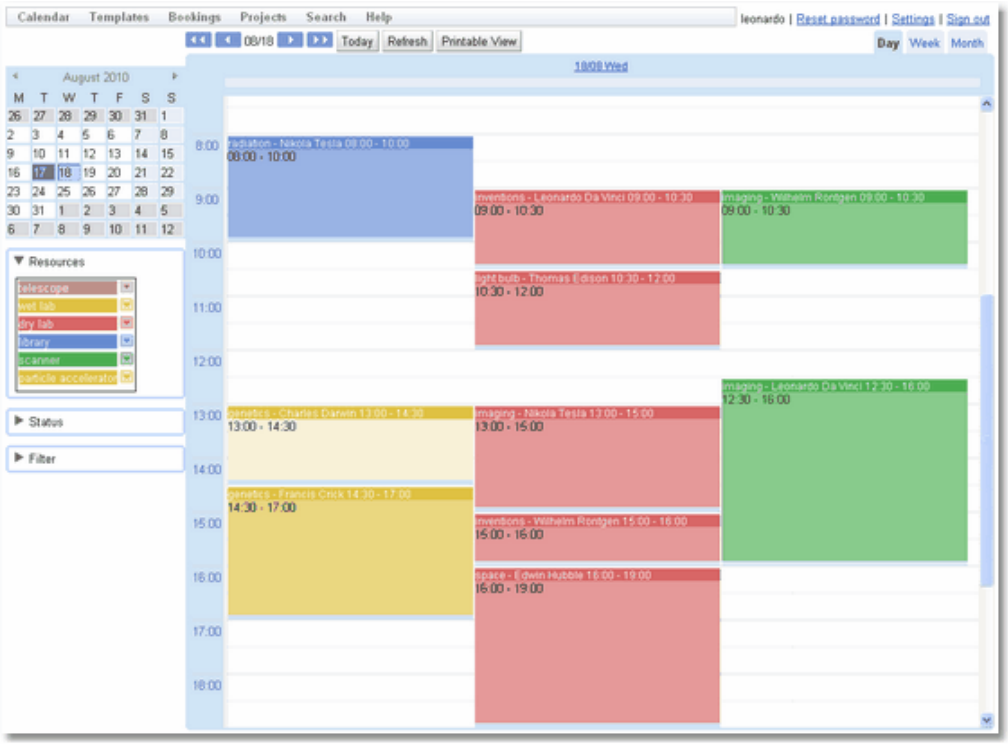
The **Refresh** button will refresh the display and find any changes that have been made since *bookings* where last fetched.

At the top of each day in the week view is a label showing the date. Click on that date to go to a day view of that date. There's also a tab selector on the far right to choose the day, week or month view.

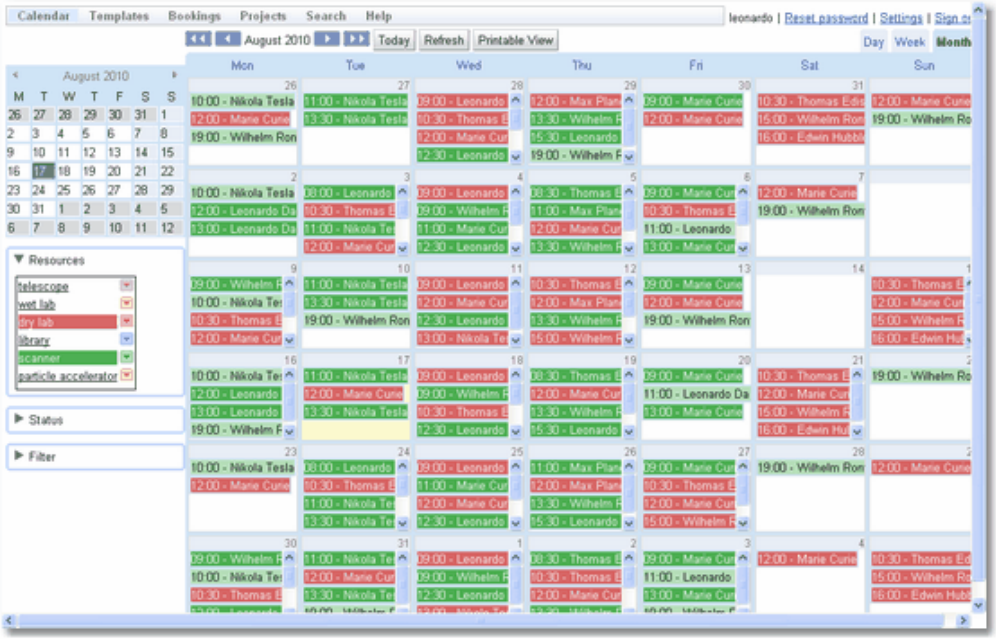


© 2010-2015 Exprodo Software

This is a day view. *Bookings* for different resources may display in the same column, as long as they don't conflict with each other. This is shown in this screen shot by the yellow and blue *bookings*.

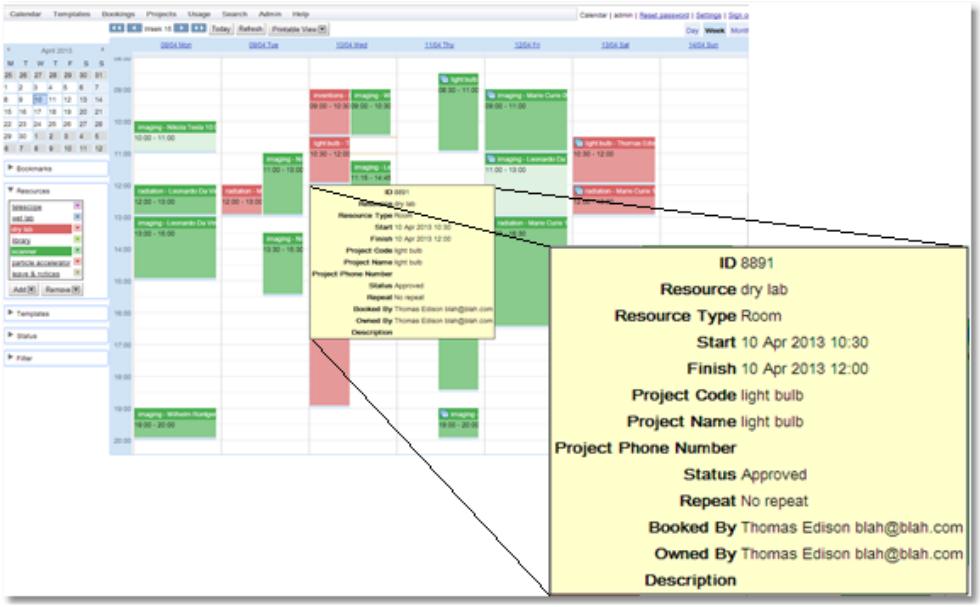


This is a month view. It can get very cluttered if *bookings* are not filtered.



© 2010-2015 Exprodo Software

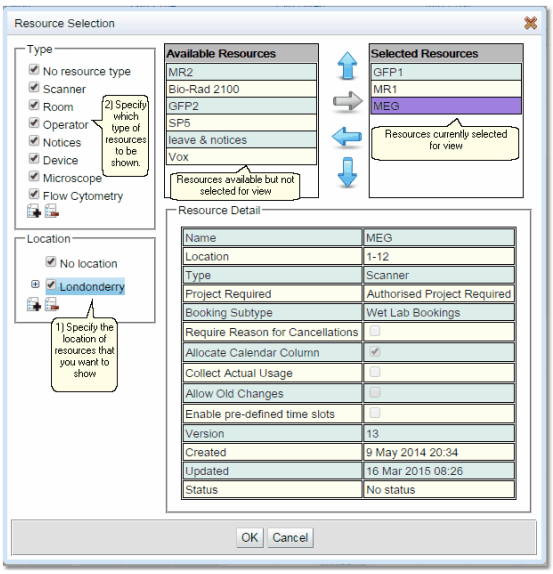
When the mouse hovers over a *booking*, a tool tip will appear which shows some of the details for the *booking*. More information can be found by clicking on the *booking* to get the *booking* pop-up, but the tool tip shows the most important information.



© 2010-2015 Exprodo Software

Resource Selection

When dealing with a number of *resources* there will be times when a user will only need to see some of the available *resources*. The **Resource Selection** pop-up allows users to determine which resources will be shown.



First of all select the location the *resources* come from. Selecting a top level location will automatically select all the lower level locations associated with it, the reverse does not happen. Lower level locations can be selected without their upper level selected.

Then select the types of *resources* required to be viewed.

All these *resources* will appear in the **Available Resources** box. Select the resources in the **Available Resources** box to be moved to the **Selected Resources** box. This can be done one at a time, or use the ctrl key to select multiple non adjacent *resources* or the shift key to select multiple adjacent *resources*. (i.e use the mouse to select the first resource, hold the ctrl key down while using the mouse to select other *resources*)

Once all required *resources* are selected, click on the blue arrow to move them to the **Selected Resources** box.

To remove *resources* from the **Selected Resources** box go through a similar procedure.

Use the up and down arrows to change the order of the resources. The resource at the top of the list will be the left most column in the calendar.

When happy with the list of **Selected Resources** click the ***OK*** button to implement the changes.

While you are selecting *resources* the **Resource Detail** box will show the details for the latest selected *resource*.

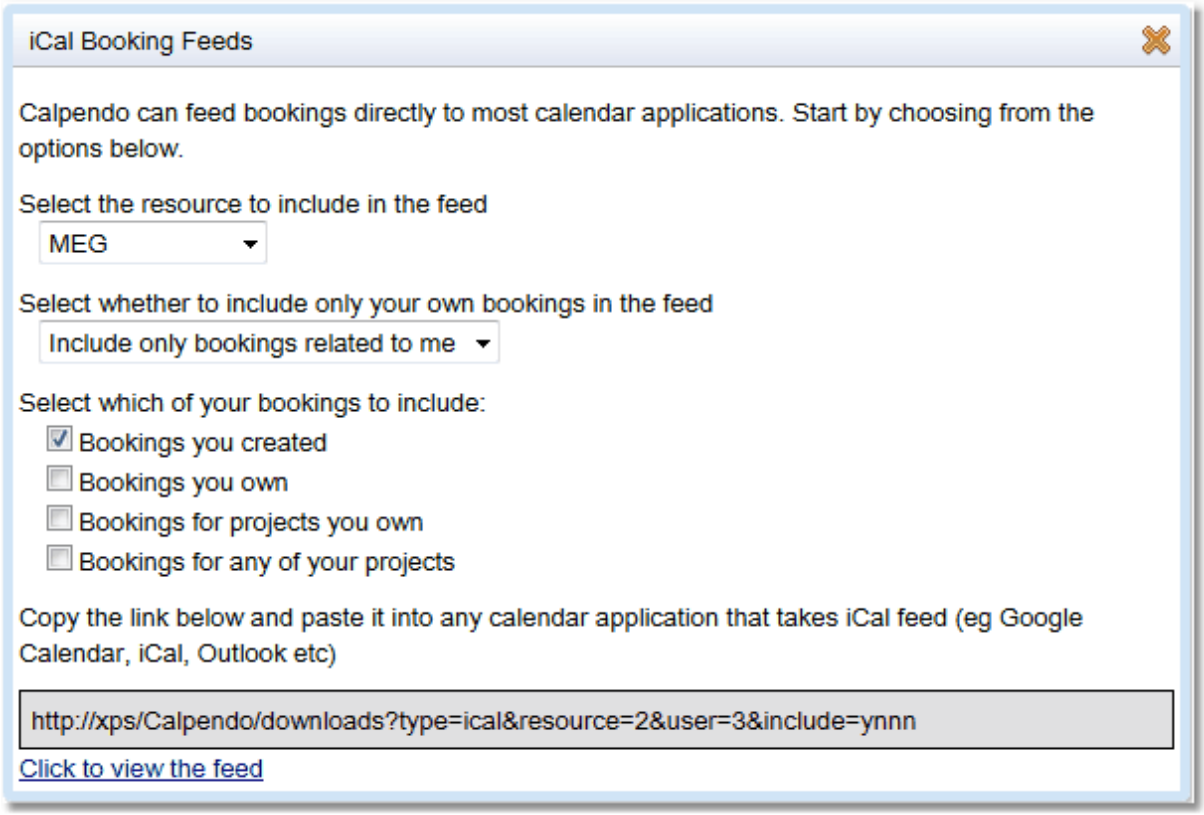
Note: If a *resource* is in the **Selected Resources** area and it is removed from that box because either its type box is unticked or its location box is unticked, then the *resource* is still a selected *resource* and will function as one later. All the user has done at this point is to remove the *resource* from view in the **Resource Selection** pop-up.

© 2010-2015 Exprodo Software

Exporting iCal Information

When exporting **iCal** information the user needs to decide if they want to export information on a single resource or all *resources*, and whether they want *bookings* related to themselves or other peoples *bookings* as well. If they want *bookings* related to themselves they can also decide which of the relationships they want to choose. Once these options have been chosen a link will appear at the bottom of the *booking* feed box, copy and paste this link to the **calendar** application. The frequency of updates to the **calendar** may be determined by the **calendar** application itself, if not, it is every hour.

If there are any **Permissions** which stop users from seeing parts of a **Booking** these may effect the **iCal** feed. For instance a **Permission** which stops **Users** from seeing the name of the **Booker** on *bookings* they have not booked will mean that all *bookings* will not be seen by the feed as when the request is made it is impossible to know who is making the request and therefore it fails.

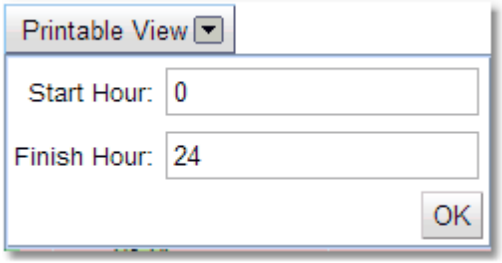

iCal Booking Feeds
Calpendo can feed bookings directly to most calendar applications. Start by choosing from the options below.
Select the resource to include in the feed
MEG
Select whether to include only your own bookings in the feed
Include only bookings related to me
Select which of your bookings to include:
Bookings you created
Bookings you own
Bookings for projects you own
Bookings for any of your projects
Copy the link below and paste it into any calendar application that takes iCal feed (eg Google Calendar, iCal, Outlook etc)
http://xps/Calpendo/downloads?type=ical&resource=2&user=3&include=ynnn
Click to view the feed

The link to be copied is made up of two parts, the first part is extracted from **Admin->General Preferences->Email->Emailed Base URL** and then the "downloads?..." section is appended. Please make sure that the base URL is set-up correctly.

© 2010-2015 Exprodo Software

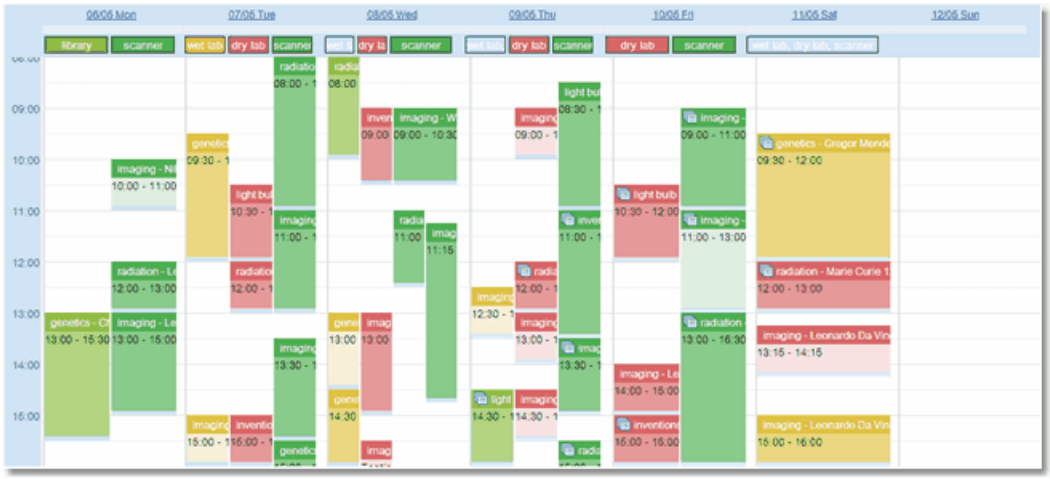
Printable View

In the *bookings calendar* you can bring up a printable view. Use the pull down to determine the hours you wish to view, and then click on the **OK** button.



The **Printable View** you get will be the same type of view as you were originally in, day, week or month, but with only the hours requested being shown. To exit the view either press the browser back button, or left click the mouse anywhere in the window.

Here is an example of a weekly view showing from 08:00-16:00.

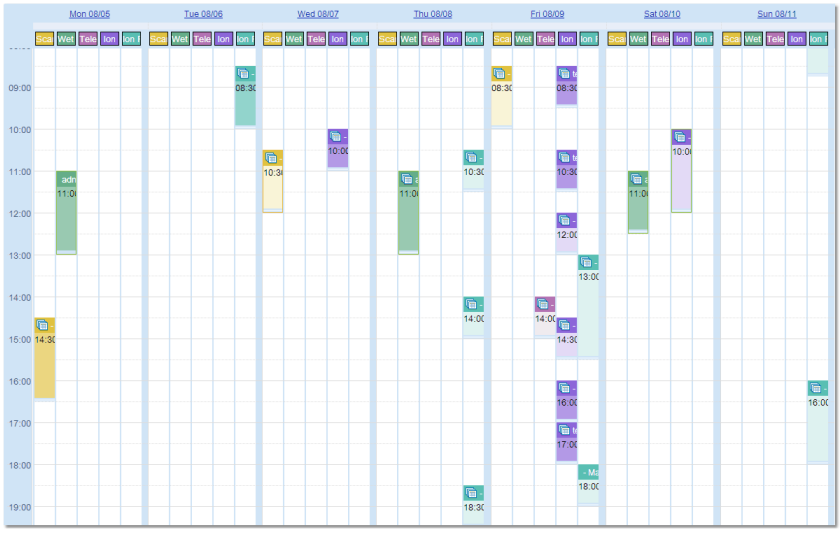


© 2010-2015 Exprodo Software

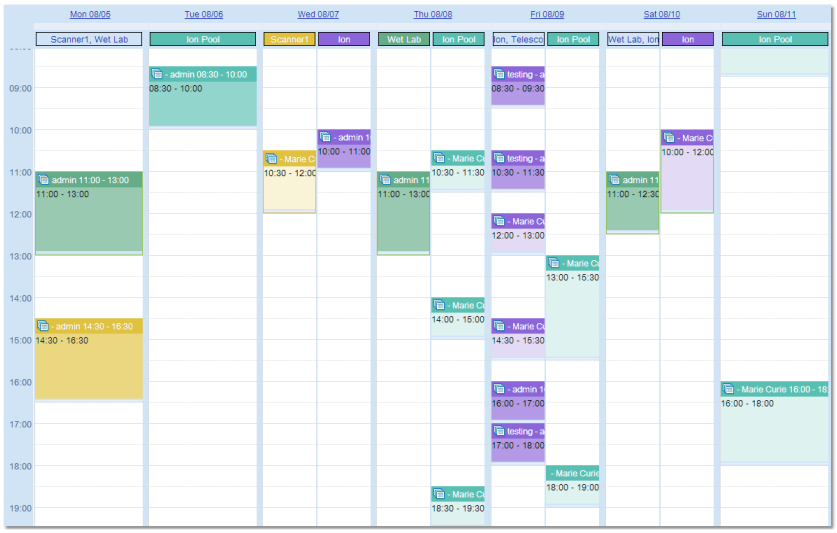
3.4.1.2 Display Of The Bookings Calendar

There are three methodologies for displaying *resources*[449] in the *calendar*[446] (changeable in a users Settings[32] or globally in Global Preferences->Bookings[331]).

1) **Always present**: with this option all *resources* will be shown in separate columns at all times. The optimal methodology for using with **Calpendo** is **Always Present**. This is because both predefined slots, and viewing *Time Templates*[450] work best. Predefined slots work best when a *resource* has its own column as the booking creation pop up can be filled in with the details of the predefined slot available at the time clicked on the calender. *Time Templates* are viewed for all *resources* easily and accurately.

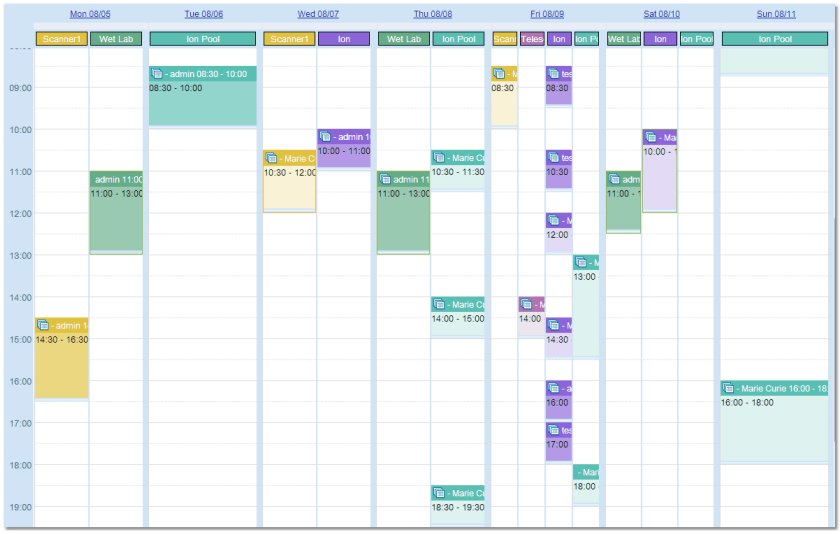


2) **Share when possible**: with this option *resources* will be displayed in the same column unless they overlap. The examples below also have their column headings switched on (also changeable in a users Settings[32] or globally in Global Preferences->Bookings[331]). Columns displaying multiple *resources* will have their headings in a neutral light blue rather than the resources colour. With the display of *Time Templates* read the section dealing with Time Templates to understand how they are displayed in shared *resource* columns. For predefined slots if there is one column or shared columns then the pop up cannot have the slots time information automatically filled in, as there is no way of knowing whether the *resource* with the predefined slots is to be used.



© 2010-2015 Exprodo Software

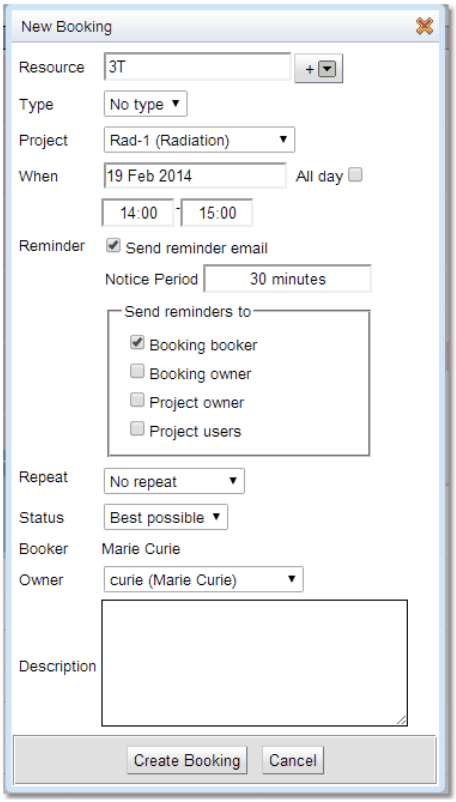
3) **Separate**: with this option all *resources* are shown in separate columns, columns are only present if the *resource* has a booking on that day. This shows *Time Templates* well, but as with **Shared when possible**, those *resources* that do not have columns will not show *Time Templates* and predefined slot information will not be filled in the create booking pop up.



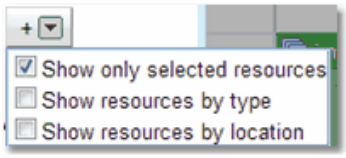
© 2010-2015 Exprodo Software

3.4.1.3 Creating Bookings

To create a new *booking*[446], in the *Bookings Calendar*[446], navigate to the date that the *booking* needs to appear on. Then either single-click in the *calendar* at the time when the *booking* is to start, or click-and-drag to select both the start and end times of the *booking*. Both methods will cause the *booking* pop-up to be displayed:



This pop-up is where the details of *bookings* are viewed and can be edited. Depending on the configuration of **Calpendo**, some of the fields displayed in this pop-up may not be editable, such as the *Status*[447] and *Owner*[448]. If the user is not allowed to make repeat[449] *bookings*, then there won't be an entry for the repeating information shown in the pop-up. Also additional fields may be present if added by the administrator.



Every *booking* must have a *resource*[449], so select which *resource* to make a *booking* for. If the *bookings calendar* had only one *resource* displayed, or the *resources* are displayed in separate columns, then the *resource* will have been automatically selected, but it can still be changed to allow *booking* for some other *resource*. The default is to display the list of *resources* currently selected for display on the calendar. If this needs to be changed, click on the pop-up next to the *resource* box and choose what and how the *resources* will be displayed.

Deselect the **Show only Selected Resources** to show all *resources*.

Select **Show resources by type** to show the *resources* in a drop down by type. This option will only be available if the administrator has Configured Types and Groups[266] for the *resources*.

Select **Show resources by location** to show the *resources* in a drop down by location.

Then click back on the **Resource** box to get the new list of *resources* to be chosen from.

Note that a *booking* is made for a *resource* that's not currently being shown in the calendar, then the calendar will automatically display all *bookings* for that new *resource*.

© 2010-2015 Exprodo Software

The administrator may have set up *Booking Types*[447] in which case choose the type of *booking*, different *booking types* may be displayed in different ways on the *calendar*.

Some *resources* require a *project*[448] for all *bookings*, and other *resources* do not. If a *resource* is selected that doesn't require a *project*, then the *project* selection will be hidden. The list of *projects* shown, will only be the *projects* the user is associated with. If the user is not associated with any *projects*, then a *booking* cannot be made for a *resource* that requires *projects*. The administrator can specify that a *project* can only book a *resource* if that *project* has an entry in its *resource settings* for the relevant *resource*. this is specified in the **Resource Editor**.

The time and date of the original selection are shown, but these can be changed before saving the *booking*. A *booking* that is set up to be **All day** will display at the top of the day column making it ideal for holiday notifications etc. If predefined *booking* slots are being used then the time in the pop up will match the slot available where the *calendar* was clicked. The times can be changed, but may not be allowed to be saved due to the **Predefined Booking Rule**.

Now set whether to get an email reminder[52] and whether the *booking* is repeatable[53]. See below for more information on how to set these up.

The *status* will default to *Best Possible*[446]. This means that the best possible outcome will be attempted from *Approved*[446], *Requested*[449] or **Denied**. If there is no *Time Template* available and the user is not denied by a *Booking Rule* or *Permission*, the Default Booking Status[331] set up in Global Preferences[328] by the administrator will be used. This choice can be overridden by using the drop down menu, although this choice may be overridden in turn by *Time Templates, Booking Rules*[446] or *Permissions*[448]. If there is a *Time Template* available and it issues a **Warning** or the booking is deemed *Acceptable*[446] then *Best Possible* becomes *Requested*. For more information on how *Time Templates* restrict bookings read the section on Time Templates[57] in the Booking Restrictions[57] chapter.

© 2010-2015 Exprodo Software

The table below will show for each type of possible *Booking Status* and *Time Template* combination for a standard user, what final *Booking Status* will be applied to the *booking*. These results may be altered by any *Booking Rules* and *Permissions* that affect *bookings*.

Initial Chosen Booking Status	Time Template Available	Final Booking Status
Best Possible	No Time Template	Default Booking Status 331
	Warning Template	Requested
	Acceptable	Requested
	Auto-Approval	Approved
Requested	No Template	Requested
	Warning Template	Requested
	Acceptable	Requested
	Auto-Approval	Approved
Approved	No Template	Permission denied
	Warning Template	Permission denied
	Acceptable	Permission denied
	Auto-Approval	Approved

The *description* field is a place for you to enter any text that describes your *booking* or is required as a comment.

Notice that the **Create Booking** button in the previous diagram is greyed out. This is because every *booking* must have a *resource*, and there is no *resource* selected yet. Similarly, if a *resource* is selected that requires a *project*, but a *project* is not selected, then the **Create Booking** button will be greyed out. **Calpendo** may also detect that the selected values are prohibited by the *Permissions* that have been configured. If this happens, then again, the **Create Booking** button will be greyed out.

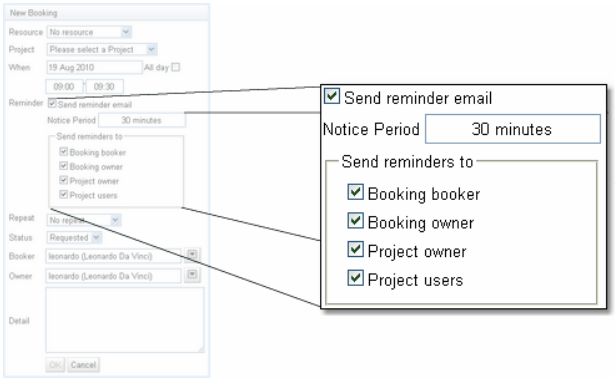
Once all the details have been entered and **Create Booking** has been pressed, then the new *booking* will be sent to the **Calpendo** server. Once there, it will undergo more tests to make sure that the *booking* is valid. An error message will appear if it is not allowed, a warning message if it is allowed, or the *booking* will be quietly accepted. Use **Cancel** to cancel the booking process.
Bookings that are created may be made as requests, or they may be auto-approved. This depends upon the configuration of the **Calpendo** and is described further in Booking Restrictions 57.

Bookings can be copied and pasted, for fast entry of similar *bookings*. Just click on a *booking* in the *calendar* and select **Copy** from the pop up menu. Then, when the *calendar* is clicked somewhere to create a new *booking*, if there is a *booking* in the clipboard, its content will be used to populate the *booking* pop-up. It can then be edited as required.

© 2010-2015 Exprodo Software

Booking Reminders

Calpendo can send an email to remind about *bookings* that have been made. Tick the **Send reminder email** box in the *booking* pop-up to see the reminder options. The default reminder options are taken from the personal booking reminder[32] settings.



The notice period indicates how far in advance of the booking the email is to be sent.

Then choose who gets a reminder email, with the options being:

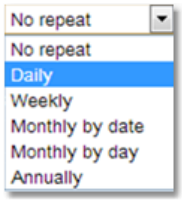
- The person that made the *booking.*
- The person that owns the *booking*.
- The person that owns the *booking's project*. If there is no *project* on the *booking*, then this setting is ignored.
- The people associated with the *booking's project*. If there is no *project* on the *booking*, then this setting is ignored.

If a person falls into more than one category they will only receive one reminder email, for example the *project* owner is also the person that made the *booking*. Reminders are not sent to users who can no longer login unless their status is set to **Lurker**.

© 2010-2015 Exprodo Software

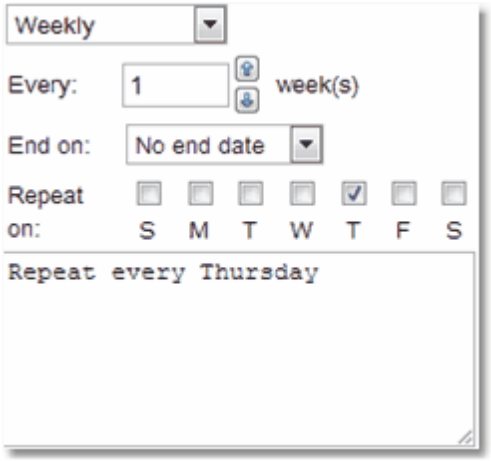
Repeat Bookings

If repeat *bookings* are allowed, then the *Repeat* option will be viewable. Select the type of *repeat* and then set the configuration options for that *repeat* type.



When creating a *booking* it can be set up to automatically *repeat*. There are a number of *repeat* choices:

Daily: Repeats at the same time each day.



Weekly: Repeats each week on this day at the same time. Specify which days the *repeat* will occur on by selected the appropriate tick boxes.

Monthly by date: Repeats each month on a particular date.

Monthly by day: Repeats each month on a particular day, this could be defined for example as: "Repeat on the 4th Monday of every month"

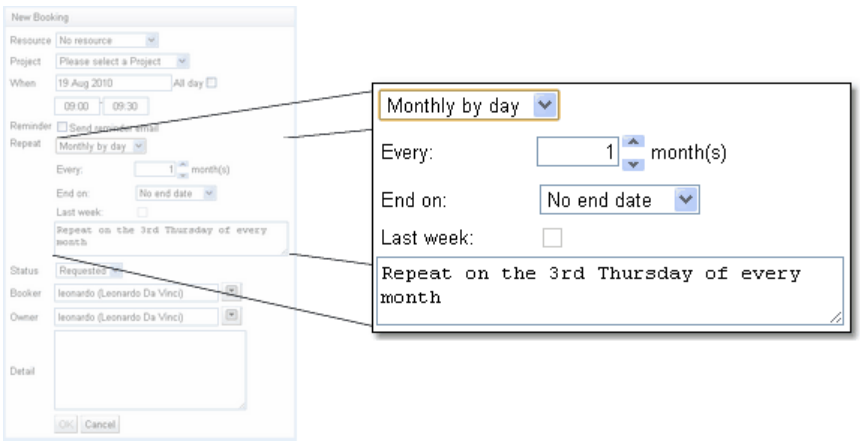
Annually: Repeats on the same date each year.

Once the type of *repeat* required has been selected, then choose how often the *repeat* occurs, every month, every other month etc.

Also define an end date for the repeated *bookings*.

If the repeat is **Monthly by day** and a day towards the end of the month has been chosen, the **Last week** enables swapping between the two possible options.
e.g. "Repeat on the 4th Monday of each month" and "Repeat on the last Monday of each month".

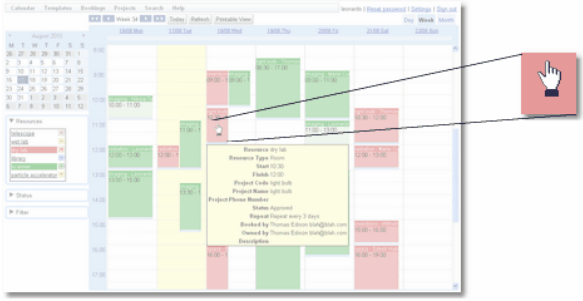
The description box at the bottom gives a written definition of the *repeat* that has been defined allowing the choices to be checked to make sure they are correct.



© 2010-2015 Exprodo Software

3.4.1.4 Editing And Cancelling Bookings

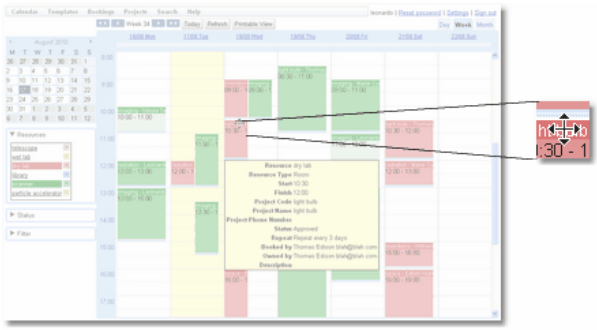
As you move the cursor over *bookings*[446] in the *calendar*[446], notice that the cursor change shape. The cursor will gives a hint about whether or not a *booking* can be edited; if a *booking*, cannot be edited then the cursor does not change as it moves over the *booking*.



This screen shot shows what the cursor will look like when the mouse moves over a *booking* that can be edited. Click with the mouse while over a *booking*, then the *booking* pop-up will appear.

If the *booking* is editable, then the fields in the pop-up will allow their values to be changed, just like creating a *booking*.

If there is no *permission*[448] to edit a *booking*, then clicking will still get the *booking* pop-up, but nothing will be able to be changed.

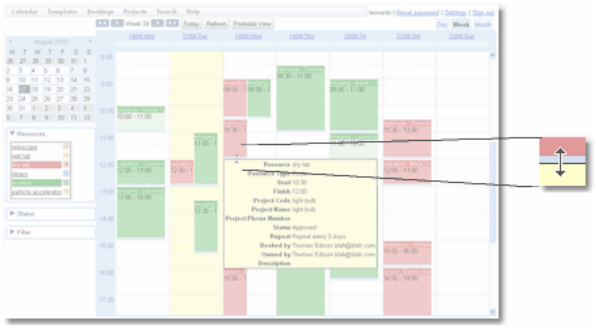


If the cursor is placed over the title bar of a *booking* that can be modified, then the cursor changes to a new shape. This indicates not only that the user can edit the *booking* by clicking to display the *booking* pop-up, but that the user can also use drag-and-drop to move the *booking*.

There may be limitations that have been set up to control where a *booking* can be dragged to, but if there is *permission*, then a *booking* can be dragged to a different start time or different day.

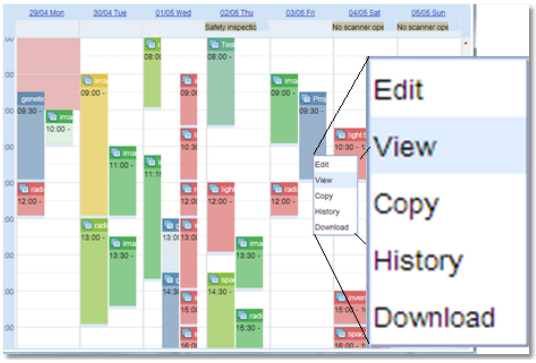
Dragging weekly *repeatable*[449] *bookings* to different days from their original *booking* day will cause the *booking* to move unpredictably. To change this type of *booking*, edit the *booking* and select different days of the week for the *booking* to be on.

© 2010-2015 Exprodo Software



Finally, by placing the cursor over the very bottom of a *booking*, the cursor changes to indicate that the *booking* duration can be changed by drag-and-drop. Click on the re-size bar at the bottom of a *booking* and drag it to the required time for the *booking* to end.

When a *booking* is dragged and dropped, or its duration changed by dragging its re-size bar at the bottom, then the change is made without showing the *booking* pop-up. However, by clicking on a *booking*, a pop-up menu is shown:



Edit : Will bring up the **Edit Booking** pop up as seen below, this is almost identical to the one seen when creating a new *booking*.

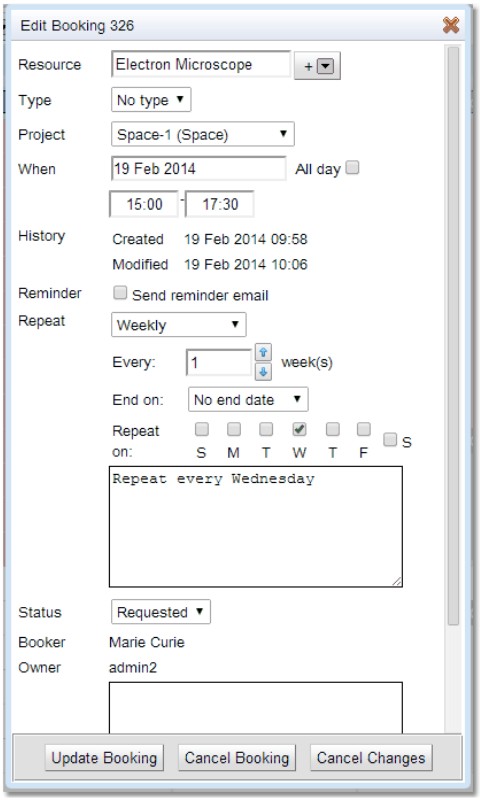
View: Will bring up a similar pop up but everything will be read only.

Copy: Will put the *booking* in the clipboard, when the user clicks somewhere to create a new *booking*, if there is a *booking* in the clipboard, its content will be used to populate the *booking* pop-up.

History: This will display the history of the *booking* showing when it was created and any updates to the *booking*. A *repeatable booking* will have one update for each repeat.

Download: Downloads the *booking* to your **Outlook** calendar.

© 2010-2015 Exprodo Software



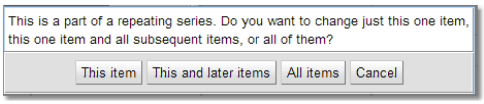
The title of the pop-up now says **Edit Booking** instead of **New Booking**.

Some of the fields may not be changeable. For example, there's a **History** field that shows when the *booking* was created or last modified and a user will probably not have *permission* to change those values. Once changes have been completed use the **Update Booking** to implement the changes. Use the **History** from the pop up described above to get a full history of changes to this *booking*.

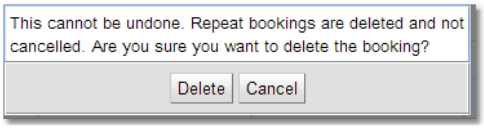
There is also a **Cancel Booking** button. *Bookings* are never deleted, they are only cancelled (unless they are *repeatable bookings*, see below). This means that the status on a *booking* is changed to **Cancelled**. **Cancelled** *bookings* are not normally displayed on the *Bookings Calendar*, so a **Cancelled** *booking* would disappear from the calendar. However, you may choose to display **Cancelled** *bookings*. **Calpendo** may be configured so that cancelling a *booking* for a particular *resource* requires a reason.

It is possible to uncancel a *booking* by displaying **Cancelled** *bookings* and then changing its status to *Requested*[449] or *Approved*[446], but suitable *Permissions* would be needed to be able to do that.

Use the **Cancel Changes** button to remove any edits and to exit the pop up.If a *repeatable booking* is being edited, any occurrence of this *booking* in the past will not be affected as they are now separate unique instances, only occurrences of the *booking* in the future will be affected. This means a *repeatable booking* can be changed without affecting any information for reports of those *bookings* in the past. When dealing with *repeatable bookings* these are deleted rather than cancelled.


This is a part of a repeating series. Do you want to change just this one item, this one item and all subsequent items, or all of them?
This item
This and later items
All items
Cancel

With *repeatable bookings* an option appears asking which items to be deleted.
This Item: Just the selected *booking*.
This and later items: The selected *booking* and all *bookings* after it in the repeatable sequence.
All Items*:* The selected *booking* and all *bookings* before and after it excluding any *bookings* in the past. ie. before the current time and date. If the first *booking* of a *repeat* has been chosen then **All Items** and **This and later items** are identical in function.


This cannot be undone. Repeat bookings are deleted and not cancelled. Are you sure you want to delete the booking?
Delete
Cancel

Then a reminder that *repeat bookings* are deleted and therefore cannot be recovered.

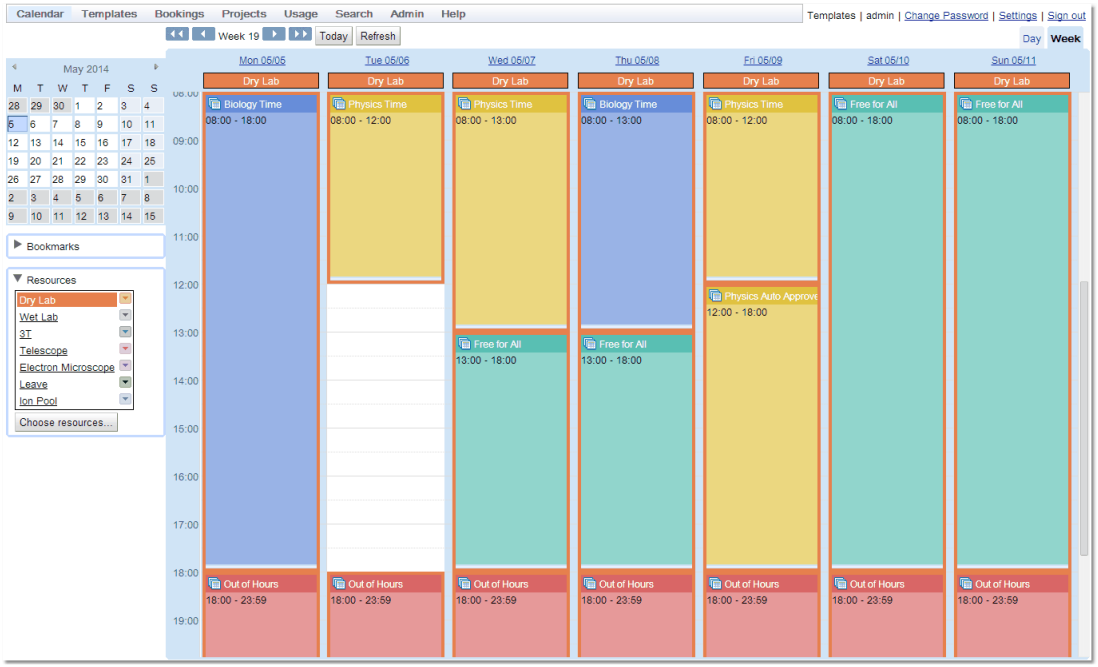
© 2010-2015 Exprodo Software

3.4.2 Booking Restrictions

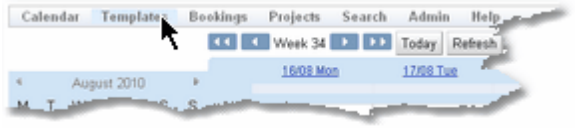
Calpendo allows users to make *bookings*[446] for themselves, while the system makes sure that *bookings* can only be created that adhere to the policies of the facility. This means that **Calpendo** will restrict what can be booked, when and for how long. These restrictions come in three types: *Time Templates*[450], booking *Booking Rules*[446] and *Permissions*[448]. If a *booking* does not pass any one of these three restrictions it will be rejected.

3.4.2.1 Time Templates

Time Templates[450] are a means of specifying periods of time during which different groups of people may book each *resource*[449]. The *Time Templates Calendar*[450] looks very much like the *Bookings Calendar*[36]. The main difference is that colour no longer pertains to the *resource*. Instead, a *Time Template* specifies who can book, and that *Time Template* spans various periods of time. Each *Time Template* is given a colour, and so the colour is an indication of who can book.



By default, the *Time Templates Calendar* appears on the **Calpendo** menu here:

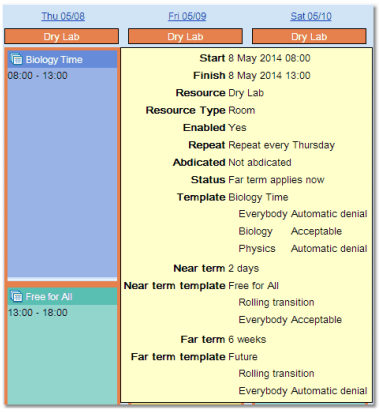


However, the administrator may have configured **Calpendo** so that the menu is different.

A *Time Template* can choose one of four possible states for a *booking*[446] request:

1. *Automatic Approval*[446]
2. *Acceptable*[446], *bookings* will be given the status *Requested*[449].
3. *Warning*, *bookings* will be given the status *Requested* but a warning message will be issued.
4. *Automatic Denial*[446]

© 2010-2015 Exprodo Software



Hovering over the blue *Time Template* for Thursday morning, which is named **Biology Time**, shows a tooltip with the details indicating it is acceptable for **Biology** people to make a *booking* for the dry lab. This means that *booking* requests can be made.

It can also be seen that for **Physics** people and everybody else, *bookings* are *automatically denied*.

All blue items shown in this *Time Templates calendar* are for the same *Time Template*, and so the same restrictions apply.

Abdicated[446]: This facility allows users to give up their saved time slots if they know no one will be using the *resource*. This means that the standard *Time Template* will now use the *Near Term Time Template*[448] definition. Only users with the appropriate *permission*[448] can abdicate a time slot and these will be set up by the administrator.

Status: This indicates which of the three possible *Time Templates* are currently being used for this specific time slot. They are:

Template: The standard *Time Template* with information on the restrictions enforced by the template.

Near term: A different *Time Template* used due to the proximity of the time slot to the current data and time, defined is when the *Near Term* starts.

Near term template: These define the *Time Templates* to be used, what time period defines near or far and the *booking* acceptability for different groups and how the transition is made.

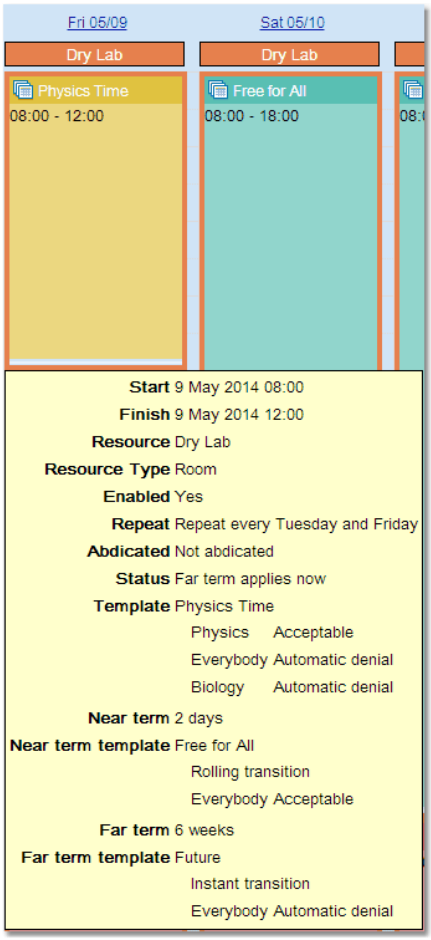
Far term[447]: A different *Time Template* used due to the need to restrict *bookings* too far in the future.

Far term template: These define the *Time Templates* to be used, what time period defines near or far and the *booking* acceptability for different groups and how the transition is made.

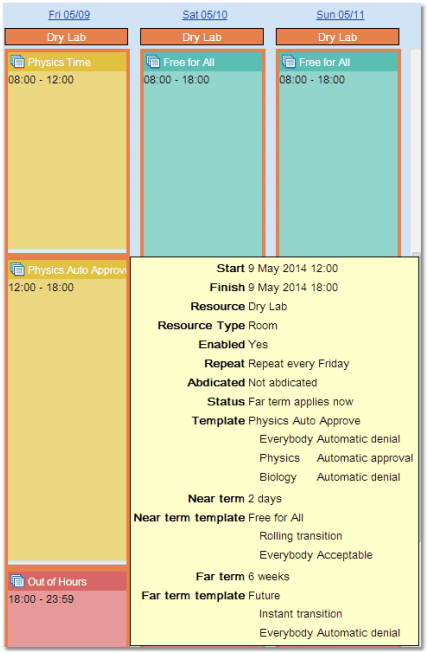
For more on *Near* and *Far Term Templates* see the appropriate section later in the chapter.

Note that the *Time Templates Calendar* does not say exactly what is meant by **Biology** and **Physics**. The names of these *Time Templates* should be chosen so that they are meaningful enough that everyone understands what they mean. The details of precisely what they do mean can be found by looking at Configuring Time Templates[215], but hopefully when a user looks at the *Time Templates* in **Calpendo**, they will understand what they mean from their name.

© 2010-2015 Exprodo Software



The Friday morning yellow *Time Template*, called **Physics Time**, indicates that **Biology** people are automatically denied *booking* the dry lab whereas **Physics** people are allowed to make *booking* requests.



The Friday afternoon *Time Template* is also yellow, the same colour as **Physics Time** and yet it is a different *Time Template* and called **Physics Auto Approve**. This is because whoever configures the *Time Templates* is free to assign whatever colours they want to each *Time Template*. Ideally, *Time Templates* for the same *resource* would be given different colours, but it's not an absolute requirement.

Physics Auto Approve specifies that **Physics** people will receive *automatic approval* for any *bookings* made for the dry lab during this time.

© 2010-2015 Exprodo Software

Near And Far Term Time Templates

Time Templates allow an administrator to specify who can use what when.

However, sometimes, it's more important to make sure that a *resource* is used by somebody rather than go unused because it couldn't be used by the "right" people. This is why *Time Templates* can be configured to mutate in the near term. For example, if by Thursday, the dry lab has no *bookings* for the following Friday morning, then perhaps the administrator would like to allow anybody from **Physics** or **Biology** to be allowed to book it. This is achieved by allowing the **Physics Time** *Time Template* to mutate to a more liberal *Time Template* within two days of the desired *booking* time. The tool tips above show not only what the *Time Templates* mean, but also what they mutate to in the near term.

It may also be important to make sure *resources* are not being booked too far in the future, in the hope they may be used. This is why *Time Templates* can be configured to mutate in the far term. For example, it may be policy that no one may book a *resource* more than six weeks in advance. This is achieved by allowing the **Physics Time** *Time Template* to mutate to a more restrictive *Time Template* if we are at least 6 weeks from the current time.

Transition allows the administrator to define how the change will take place from the Medium Term to the *Near* or *Far Term*. With **Instant** transition, at 9am, the whole of a *Time Template* from 9am to 5pm would become available for *booking*. With Rolling transition at 10am the portion of the *Time Template* from 9am to 10am would be available for booking.

3.4.2.2 Booking Rules

Whenever a *booking* [446] is created or modified, the **Calpendo** server checks the *booking* against its *Booking Rules* [446]. There are many different sorts of *Booking Rules*, these allow **Calpendo** to apply pretty much any check the administrator might want to run.

Normally, the only time *Booking Rules* will be noticed is if a *booking* is created or modified that triggers a particular *Booking Rule*. *Booking Rules* may do one of the following things:

- Accept the change without showing you any message.
- Accept the change, but give a warning message.
- Refuse the change and show an error message.

The administrator should give indicate which *Booking Rules* are being used by the facility.

The configuration of *Booking Rules* is beyond the scope of the **Calpendo User Guide**, but read the Configuring Booking Rules [223] section for more details.

3.4.2.3 Permissions

Permissions [448] are not discussed much in the **Calpendo User Guide** because the *Permissions* that exist are completely configurable for the facility by the administrator and the user should not need to know much about them. A user will be subject to whatever *Permissions* have been set up for **Calpendo**, and the only way a user will know when something is not allowed is if the system prevents them from attempting to do something (such as greying out buttons that lead to actions that cannot be performed) or by giving an error pop-up that will say **Permission Denied**.

To know more about *Permissions*, then see the Permissions [278] chapter.

© 2010-2015 Exprodo Software

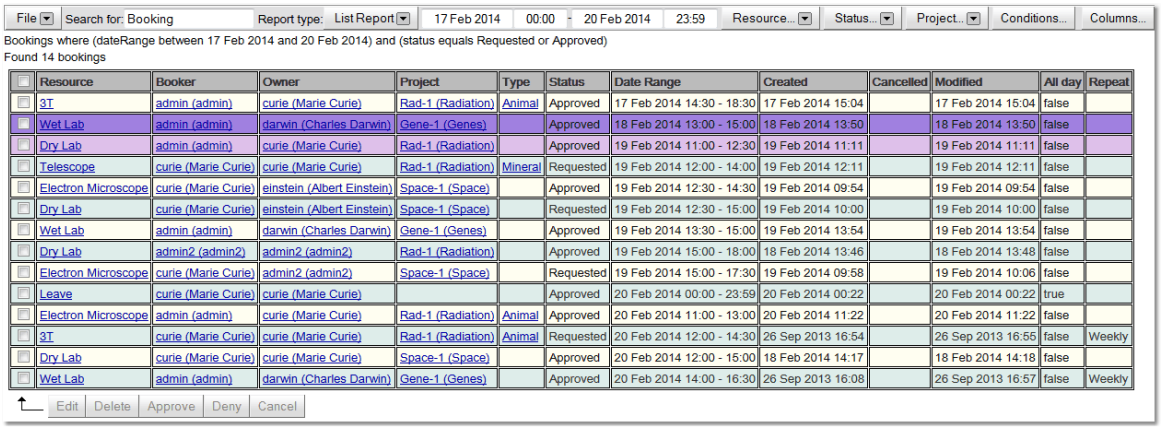
3.4.3 Searching For Bookings

Calpendo has multiple ways of finding out information about the *bookings* 446 apart from using the Bookings Calendar 36.

3.4.3.1 Using The Booking Searches

All the *booking* 446 search pages work in a similar way. This chapter explains the basic mechanisms for all the *booking* searches. To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search 108 chapter.

On entering any of the *booking* search pages a search will automatically start, of the appropriate type, with a date range of today. This is what a search will look like when it is completed.

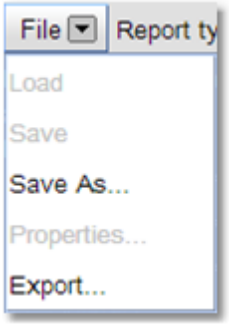

File | Search for: Booking | Report type: List Report | 17 Feb 2014 | 00:00 | 20 Feb 2014 | 23:59 | Resource... | Status... | Project... | Conditions... | Columns...

Bookings where (dateRange between 17 Feb 2014 and 20 Feb 2014) and (status equals Requested or Approved)
Found 14 bookings

Resource	Booker	Owner	Project	Type	Status	Date Range	Created	Cancelled	Modified	All day	Repeat
3T	admin (admin)	curie (Marie Curie)	Rad-1 (Radiation)	Animal	Approved	17 Feb 2014 14:30 - 18:30	17 Feb 2014 15:04		17 Feb 2014 15:04	false	
Wet Lab	admin (admin)	darwin (Charles Darwin)	Gene-1 (Genes)		Approved	18 Feb 2014 13:00 - 15:00	18 Feb 2014 13:50		18 Feb 2014 13:50	false	
Dry Lab	admin (admin)	curie (Marie Curie)	Rad-1 (Radiation)		Approved	19 Feb 2014 11:00 - 12:30	19 Feb 2014 11:11		19 Feb 2014 11:11	false	
Telescope	curie (Marie Curie)	curie (Marie Curie)	Rad-1 (Radiation)	Mineral	Requested	19 Feb 2014 12:00 - 14:00	19 Feb 2014 12:11		19 Feb 2014 12:11	false	
Electron Microscope	curie (Marie Curie)	einstein (Albert Einstein)	Space-1 (Space)		Approved	19 Feb 2014 12:30 - 14:30	19 Feb 2014 09:54		19 Feb 2014 09:54	false	
Dry Lab	curie (Marie Curie)	einstein (Albert Einstein)	Space-1 (Space)		Requested	19 Feb 2014 12:30 - 15:00	19 Feb 2014 10:00		19 Feb 2014 10:00	false	
Wet Lab	admin (admin)	darwin (Charles Darwin)	Gene-1 (Genes)		Approved	19 Feb 2014 13:30 - 15:00	19 Feb 2014 13:54		19 Feb 2014 13:54	false	
Dry Lab	admin2 (admin2)	admin2 (admin2)	Rad-1 (Radiation)		Approved	19 Feb 2014 15:00 - 18:00	18 Feb 2014 13:46		18 Feb 2014 13:48	false	
Electron Microscope	curie (Marie Curie)	admin2 (admin2)	Space-1 (Space)		Requested	19 Feb 2014 15:00 - 17:30	19 Feb 2014 09:58		19 Feb 2014 10:06	false	
Leave	curie (Marie Curie)	curie (Marie Curie)			Approved	20 Feb 2014 00:00 - 23:59	20 Feb 2014 00:22		20 Feb 2014 00:22	true	
Electron Microscope	admin (admin)	curie (Marie Curie)	Rad-1 (Radiation)	Animal	Approved	20 Feb 2014 11:00 - 13:00	20 Feb 2014 11:22		20 Feb 2014 11:22	false	
3T	curie (Marie Curie)	curie (Marie Curie)	Rad-1 (Radiation)	Animal	Requested	20 Feb 2014 12:00 - 14:30	26 Sep 2013 16:54		26 Sep 2013 16:55	false	Weekly
Dry Lab	curie (Marie Curie)	curie (Marie Curie)	Space-1 (Space)		Approved	20 Feb 2014 12:00 - 15:00	18 Feb 2014 14:17		18 Feb 2014 14:18	false	
Wet Lab	admin (admin)	darwin (Charles Darwin)	Gene-1 (Genes)		Approved	20 Feb 2014 14:00 - 16:30	26 Sep 2013 16:08		26 Sep 2013 16:57	false	Weekly

Edit | Delete | Approve | Deny | Cancel

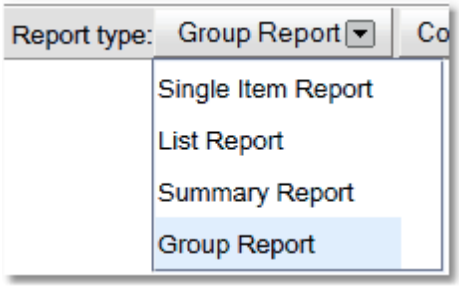
The following section goes through the options available after the search has completed, enabling the user to change the search, save it or export the data, as well as change the information viewed by the search. Not all options will be available for every *booking* search type, but that will be outlined in the specific search types chapter. For more information on any of the features shown read the Search 108 chapter.



Save As: Saves the report for later use.

Export: Export the information found in the search to a file.

© 2010-2015 Exprodo Software



This button determines what type of report will be seen:

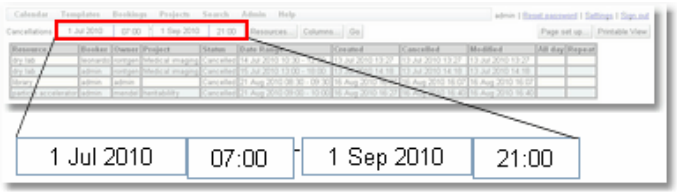
Single Item Report[449] Displays a single record that matches the search.

List Report[447]: Lists each record found that matches the search one under the other.

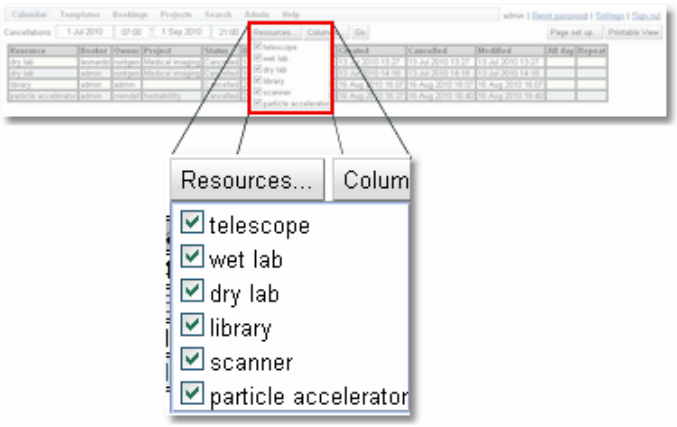
Summary Report[449]: Allows a summary of the information found.

Group Report[447]: Allows the report to be shown by groupings with a count of the number of records that fit each group.

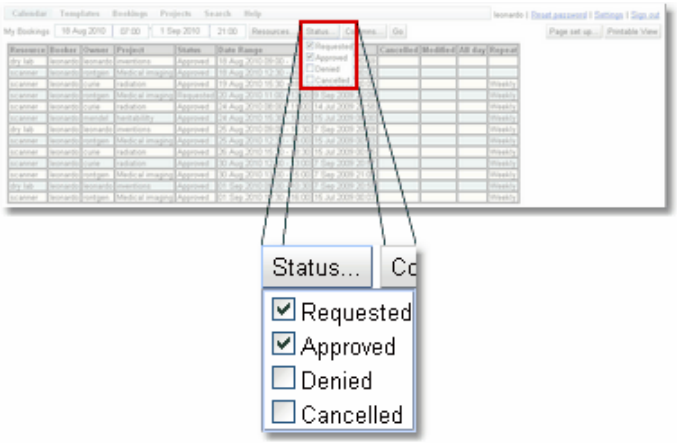
To change the search parameters



First select an appropriate range of dates for the new search.

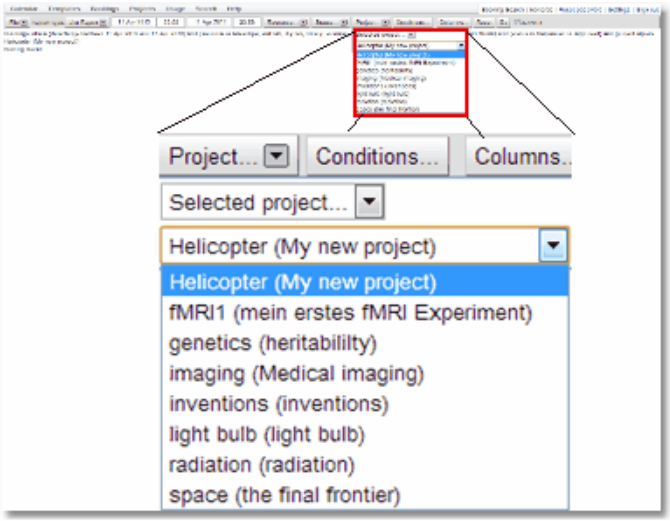


Then select the *resources*[449] required.



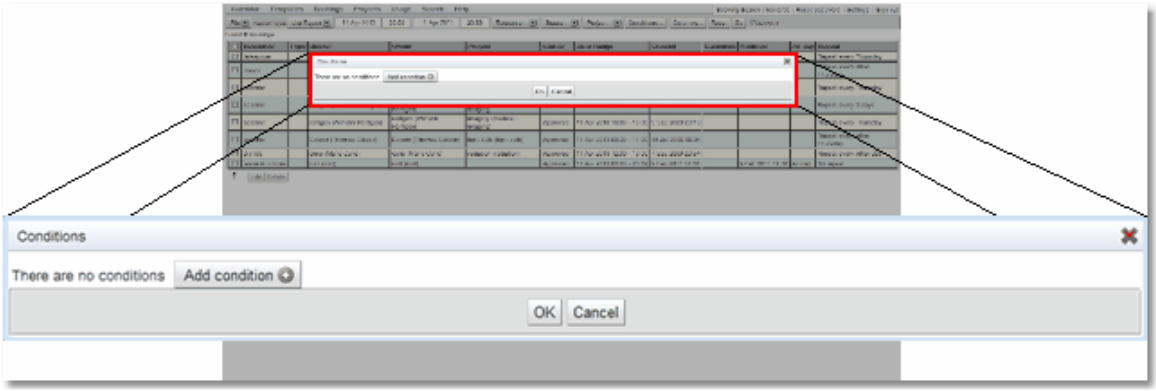
And select the required *booking statuses*[447].

© 2010-2015 Exprodo Software

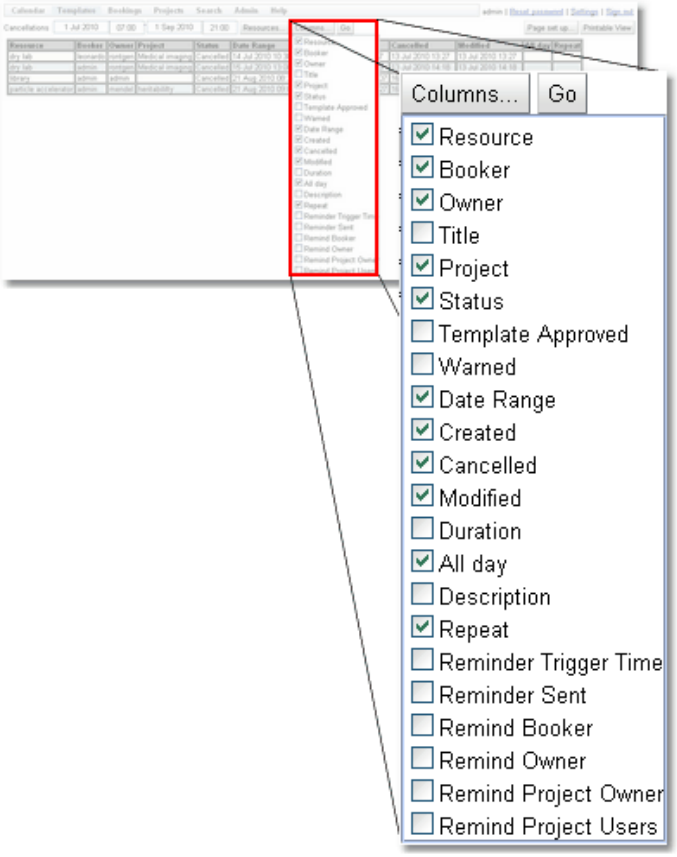


Then select the required *project*[448].

Additional *conditions*[447] for the search can also be set up. For more information on how to set up *conditions* read the information in the Setting Search Conditions page.



© 2010-2015 Exprodo Software



And then change the columns displayed in the output.

Reset: Resets the search *conditions* to the defaults set up on starting the search.

Go: Runs the search with the current *conditions*.

Autorun: If this button is ticked then searches will run as soon as any *conditions* change. If a number of the *conditions* of the search will be changed then it is more efficient if this is not ticked as each change will start a search and the appropriate transfer of data from the server.

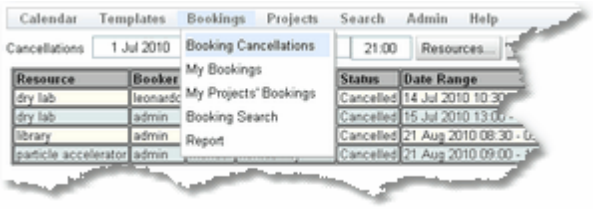
For a more complete explanation on how to edit records in a booking search view read the How to Edit Multiple Items At Once[128] section of the Data Explorer[127] chapter. The only difference is in the booking search views as well as **Edit** and **Delete** there are also options to **Approve**, **Deny** or **Cancel** multiple *bookings*.

© 2010-2015 Exprodo Software

3.4.3.2 Booking Cancellations

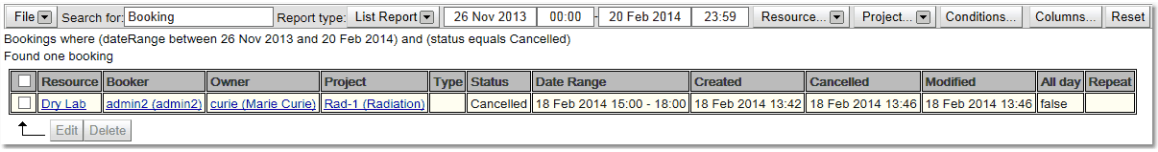
The **Booking Cancellations** search page allows searches for *bookings* [446] that have been cancelled.

By default, the **Booking Cancellations** page appears on the menu here:



However, the administrator may have configured **Calpendo** so that the menu is different.

This is what the **Booking Cancellations** page looks like once a search has been done:



The **Booking Cancellation** search is run with the following default parameters:

Bookings where (**dateRange** between **start_date** and **end_date**) and (*status* [447] equals **Cancelled**)

Because the *status* condition is set specifically to **Cancelled** there is no **Status** button available but the other buttons can be used to change the parameters of the search.

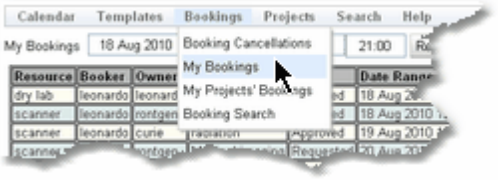
To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search [108] chapter.

© 2010-2015 Exprodo Software

3.4.3.3 My Bookings

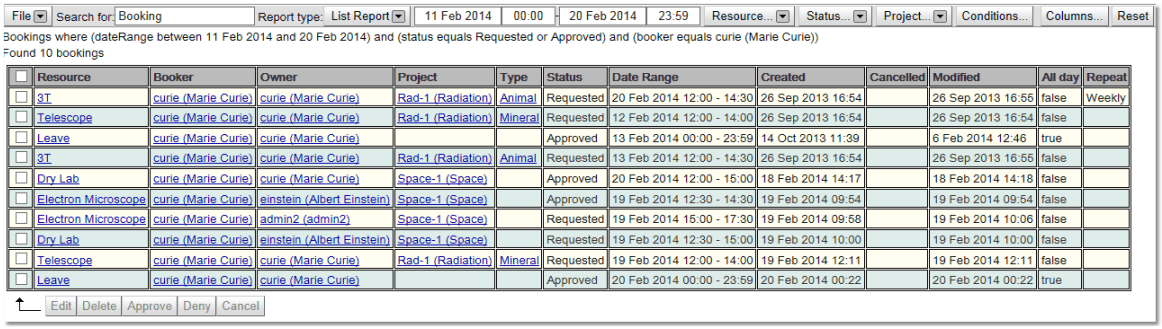
The **My Bookings** search page does a search for *bookings*[446] that the user has created.

By default, the **My Bookings** page appears on the menu here:



However, the administrator may have configured **Calpendo** so that the menu is different.

This is what the **My Bookings** page looks like once a search has been done:



Resource	Booker	Owner	Project	Type	Status	Date Range	Created	Cancelled	Modified	All day	Repeat
3T	curie (Marie Curie)	curie (Marie Curie)	Rad-1 (Radiation)	Animal	Requested	20 Feb 2014 12:00 - 14:30	26 Sep 2013 16:54		26 Sep 2013 16:55	false	Weekly
Telescope	curie (Marie Curie)	curie (Marie Curie)	Rad-1 (Radiation)	Mineral	Requested	12 Feb 2014 12:00 - 14:00	26 Sep 2013 16:54		26 Sep 2013 16:54	false	
Leave	curie (Marie Curie)	curie (Marie Curie)			Approved	13 Feb 2014 00:00 - 23:59	14 Oct 2013 11:39		6 Feb 2014 12:46	true	
3T	curie (Marie Curie)	curie (Marie Curie)	Rad-1 (Radiation)	Animal	Requested	13 Feb 2014 12:00 - 14:30	26 Sep 2013 16:54		26 Sep 2013 16:55	false	
Dry Lab	curie (Marie Curie)	curie (Marie Curie)	Space-1 (Space)		Approved	20 Feb 2014 12:00 - 15:00	18 Feb 2014 14:17		18 Feb 2014 14:18	false	
Electron Microscope	curie (Marie Curie)	einstein (Albert Einstein)	Space-1 (Space)		Approved	19 Feb 2014 12:30 - 14:30	19 Feb 2014 09:54		19 Feb 2014 09:54	false	
Electron Microscope	curie (Marie Curie)	admin2 (admin2)	Space-1 (Space)		Requested	19 Feb 2014 15:00 - 17:30	19 Feb 2014 09:58		19 Feb 2014 10:06	false	
Dry Lab	curie (Marie Curie)	einstein (Albert Einstein)	Space-1 (Space)		Requested	19 Feb 2014 12:30 - 15:00	19 Feb 2014 10:00		19 Feb 2014 10:00	false	
Telescope	curie (Marie Curie)	curie (Marie Curie)	Rad-1 (Radiation)	Mineral	Requested	19 Feb 2014 12:00 - 14:00	19 Feb 2014 12:11		19 Feb 2014 12:11	false	
Leave	curie (Marie Curie)	curie (Marie Curie)			Approved	20 Feb 2014 00:00 - 23:59	20 Feb 2014 00:22		20 Feb 2014 00:22	true	

The **My Bookings** search is run with the following default parameters:

Bookings where (**dateRange** between **start_date** and **end_date**) and (*status*[447] equals **Requested** or **Approved**) and (*booker*[446] equals **current_user**)

All the possible buttons are available to change parameters of the search.

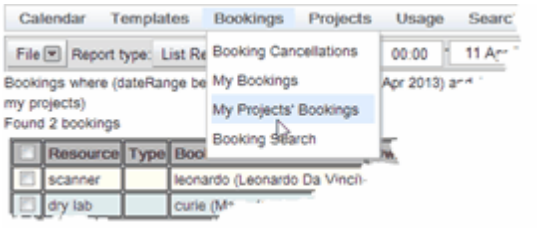
To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search[108] chapter.

© 2010-2015 Exprodo Software

3.4.3.4 My Projects' Bookings

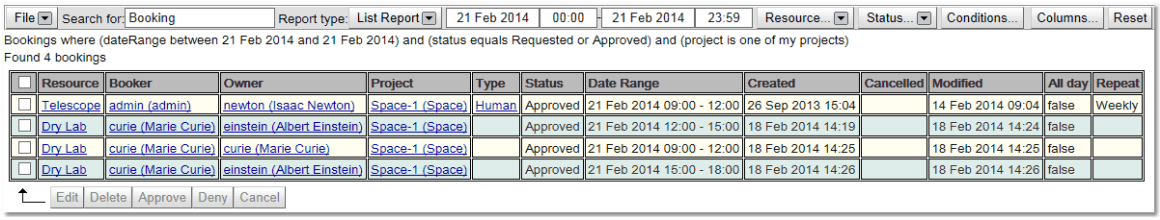
The **My Projects' Bookings** page allows searches for *bookings* 446 that use *projects* 448 that have been created by the user.

By default, the **My Projects' Bookings** page appears on the menu here:



However, the administrator may have configured **Calpendo** so that the menu is different.

This is what the **My Projects' Bookings** page looks like once a search has been done:



Resource	Booker	Owner	Project	Type	Status	Date Range	Created	Cancelled	Modified	All day	Repeat
Telescope	admin (admin)	newton (Isaac Newton)	Space-1 (Space)	Human	Approved	21 Feb 2014 09:00 - 12:00	26 Sep 2013 15:04		14 Feb 2014 09:04	false	Weekly
Dry Lab	curie (Marie Curie)	einstein (Albert Einstein)	Space-1 (Space)		Approved	21 Feb 2014 12:00 - 15:00	18 Feb 2014 14:19		18 Feb 2014 14:24	false	
Dry Lab	curie (Marie Curie)	curie (Marie Curie)	Space-1 (Space)		Approved	21 Feb 2014 09:00 - 12:00	18 Feb 2014 14:25		18 Feb 2014 14:25	false	
Dry Lab	curie (Marie Curie)	einstein (Albert Einstein)	Space-1 (Space)		Approved	21 Feb 2014 15:00 - 18:00	18 Feb 2014 14:26		18 Feb 2014 14:26	false	

The **My Projects' Booking** search is run with the following default parameters:

Bookings where (**dateRange** between **start_date** and **end_date**) and (*status* 447 equals **Requested** or **Approved**) and (*project* is **one of my projects**)

Because the *project* parameter is set specifically to **one of my projects** there is no **Project** button available but the other buttons can be used to change the parameters of the search.

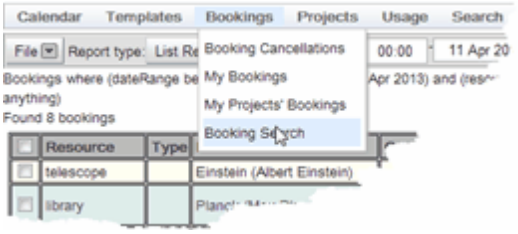
To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search 108 chapter.

© 2010-2015 Exprodo Software

3.4.3.5 Booking Search

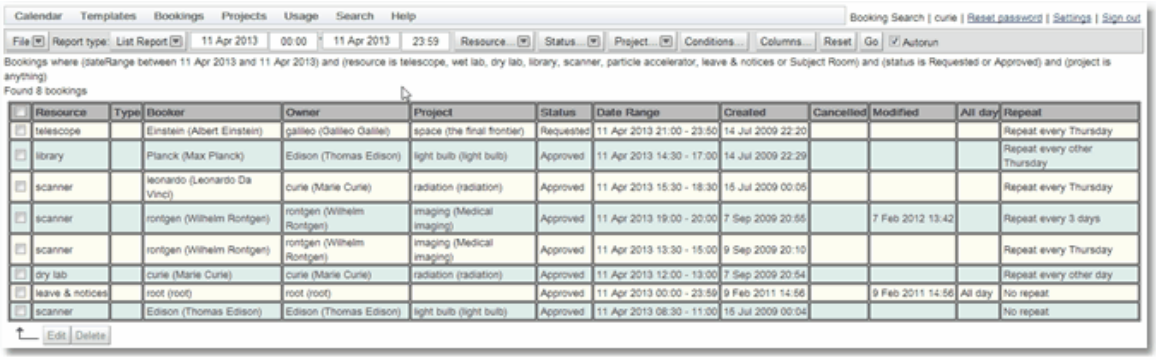
The **Booking Search** page sets up a search for all *bookings*[446] for a certain date range, it defaults to a date range which specifies today.

By default, the **Booking Search** page appears on the menu here:



However, the administrator may have configured **Calpendo** so that the menu is different.

This is what the **Booking Search** page looks like once a search has been done:



The **Booking** search is run with the following default parameters:

Bookings where (**dateRange** between **start_date** and **end_date**) and (*status*[447] is **Requested** or **Approved**)

All the possible buttons are available to change parameters of the search.

To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search[108] chapter.

© 2010-2015 Exprodo Software

3.5 Projects

Calpendo allows management of *projects*[448] that can be associated with *bookings*[446]. This helps keep track of what the *resources*[449] are being used for, but can also be used as a means of controlling who can book which *resources*. A *project* can have multiple users associated with it, and each user may only book for those *projects* that they are associated with.

The information stored within a *project* can be customised for the facility, as described in Project Configuration[205]. The screen shots that are shown in this section are likely to be different from those that the user in your facility would see, depending on how the *projects* have been set up.

© 2010-2015 Exprodo Software

3.5.1 Creating Projects

To create a new *project*[448], go to the menu option **Projects-->Create Project**.

The content viewed when creating a new *project* will vary depending on how the systems administrator has configured **Calpendo**. By default each *project* contains the following:

Property	Description
Project Code	This is an identifier that will be assigned by an administrator. These should be unique, but *projects* may not have a *project code* assigned to them until the *project approval process*[448] has completed.
Type	The facility may, or may not, classify *projects* by *type*[448]. If types are configured for use, then the user will be able to select the *type* that should be used for your *project*.
Status	Newly created *projects* will have a *status* of **Requested**. The user will probably not be able to modify the *status* of their own *projects*, but this depends on how the *Permissions*[448] are configured to allow access.
Owner	The owner of a *project* is normally the person that created the *project* request. The user may be able to select a different owner, but the administrator may also reassign the ownership of a *project*.
Name	This should be a *short* text description of the *project*. The maximum length that this can be will have been configured by the administrator.
Description	This is a longer description of your *project*.
Phone Number	Phone number to be associated with the *project*.
Start	Start date of the *project*.
Finish	Finish date of the *project*.
Principle Investigator	The main investigator of the *project*.
Ethics Approval Number	The Ethics Approval Number for the *Project*.
Funding Agency	Name of the funding agency.
Account Number	Account number for the funding.
Project Resource Settings	This specifies the *resources*[449] that the *project* can book, and provides a place to store some information relating to what will be used. For example, this may record how much the *project* will be charged for using each *resource* or how much time in total is allowed to be booked for the *project*.
Project Service Settings	This specifies the services that the *project* can order, and provides a place to store some information relating to what will be used. For example, this may record how much the *project* will be charged for ordering each service.
Users	This specifies all the users that are allowed to make *bookings*[446] on behalf of the *project*.

Once the relevant information has been filled in, press the **Submit project request** button.

© 2010-2015 Exprodo Software

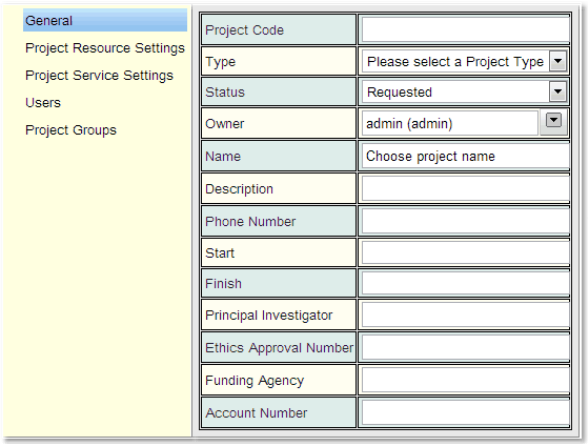
The administrator may have added a number of other pieces of information that need to be completed for a *project* to be created. If there are problems filling out the *project* creation form please talk to the **Calpendo** administrator.

The process of approving *projects* can vary a great deal between different **Calpendo** installations. The administrator should be able to tell the user about the process that's in place at the facility. However, in all cases, a *project* becomes available for booking once its status is changed to **Approved**.

See The Project Approval Process 145 for information on the options for approving *projects*.

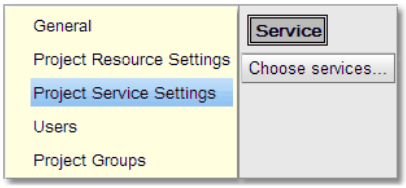
As soon as a *project* request has been created, it will appear on the My Projects 76 page.

Here is an example of creating a *project*, click on **Create Project**,



Project creation starts with the **General** tab. Fill in the form and then move on to the other tabs.

Click on **Choose Resource** to get a pop up which will allow the choice of *resources* to be used for this *project*. View the section on Resource Selection to see how to select *resources* using this pop up.



Click on **Project Service Settings** to get a pop up which will allow the project manager to choose the services available to be ordered by this project.

The current user name will appear automatically in the user list. Use the drop down to select other users to be added to the list.

To remove a user, tick the option box and click **Remove**.

Now submit the *project* request using the **Submit project request** button.

The **Project Groups** tab cannot be used until the *project* is initially saved.

© 2010-2015 Exprodo Software

3.5.2 Modifying Projects

Modification of *projects*[448] is done using the My Projects[76] and Project Search[72] pages. Each of those pages shows a list of *projects*, and allows the selection of a particular *project* so that its full details can be seen, and also to be able to edit it. See the documentation for My Projects[76] and Project Search[72] for full details. Note that the ability to edit a *project* is always subject to the user having *permission*[448] to do so.

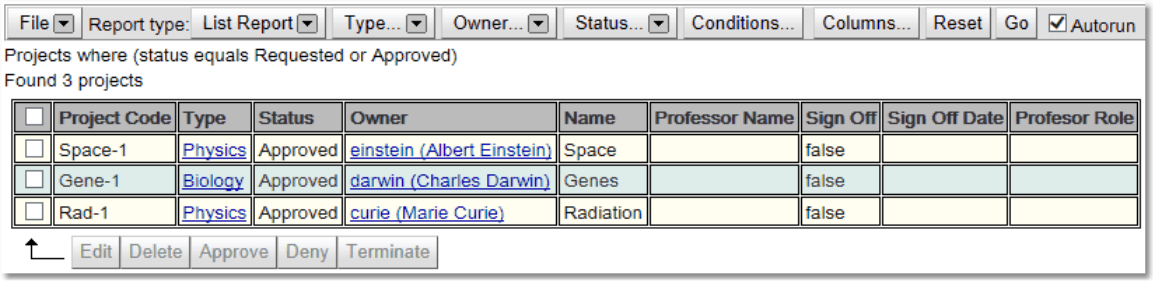
3.5.3 Searching For Projects

Calpendo has multiple ways finding out information about the *projects*[448] that are being used.

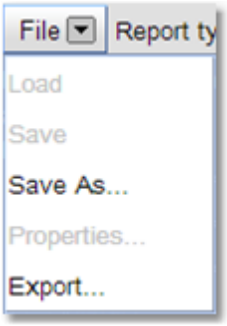
3.5.3.1 Using The Project Searches

All the *project*[448] search pages work in a similar way. This chapter explains the basic mechanisms for all the *project* searches. To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search[108] chapter.

On entering any of the *project* search pages **Calpendo** will automatically start a search of the appropriate type with a date range of today. This is what a search will look like when it is completed.


File
Report type: List Report
Type...
Owner...
Status...
Conditions...
Columns...
Reset
Go
Autorun
Projects where (status equals Requested or Approved)
Found 3 projects
Project Code | Type | Status | Owner | Name | Professor Name | Sign Off | Sign Off Date | Profesor Role
Space-1 | Physics | Approved | einstein (Albert Einstein) | Space | | false | |
Gene-1 | Biology | Approved | darwin (Charles Darwin) | Genes | | false | |
Rad-1 | Physics | Approved | curie (Marie Curie) | Radiation | | false | |
Edit
Delete
Approve
Deny
Terminate

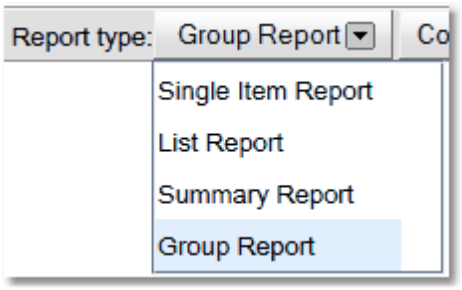
The following section goes through the options available after the search has completed, enabling the user to change the search, save it or export the data, as well as change the information viewed by the search. Not all options will be available for every *project* search type, but that will be outlined in the specific search types chapter. For more information on any of the features shown you will need to read the Search[108] chapter.



Save As: Saves the created report for later use.

Export: Exports the information found in the search to a file.

© 2010-2015 Exprodo Software



This button determines which type of report will be seen:

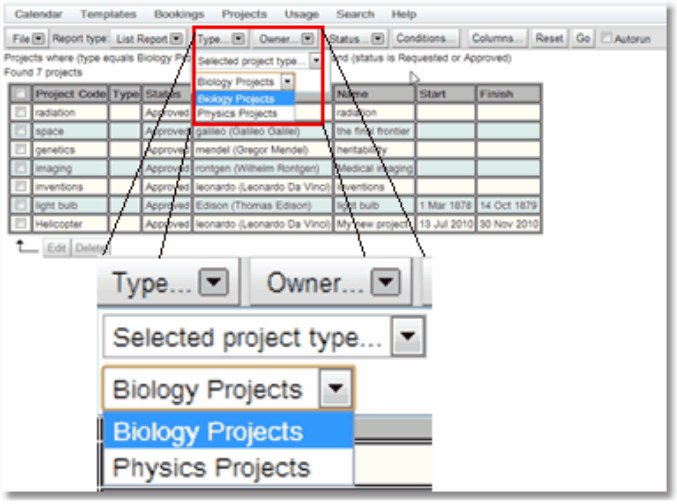
Single Item Report[449] Displays a single record that matches the search.

List Report[447]: Lists each record found that matches the search one under the other.

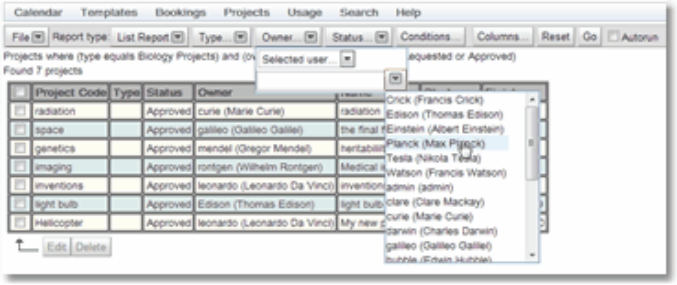
Summary Report[449]: Summarises the information found.

Group Report[447]: Shows the report by groupings with a count of the number of records that fit each group.

To change the search parameters

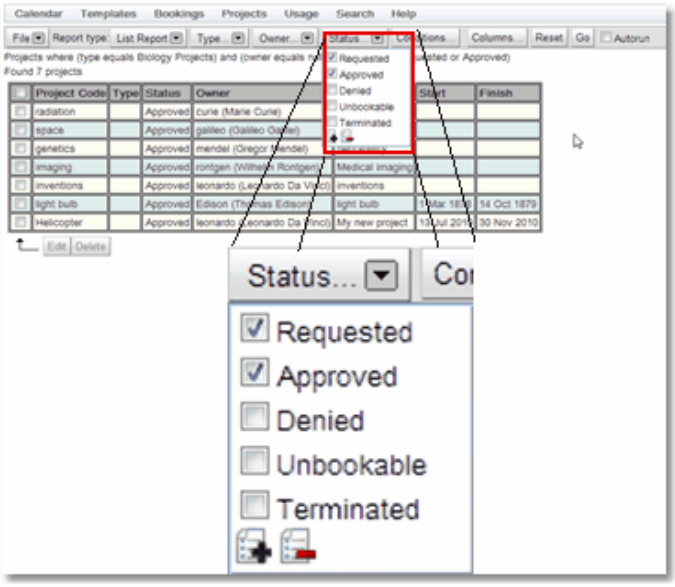


First select the type of *project* required



Then select the owner needed.

© 2010-2015 Exprodo Software



And select the required *project statuses*[448].
Requested: Requested *projects* are those that are still going through the *project approval process*[448].

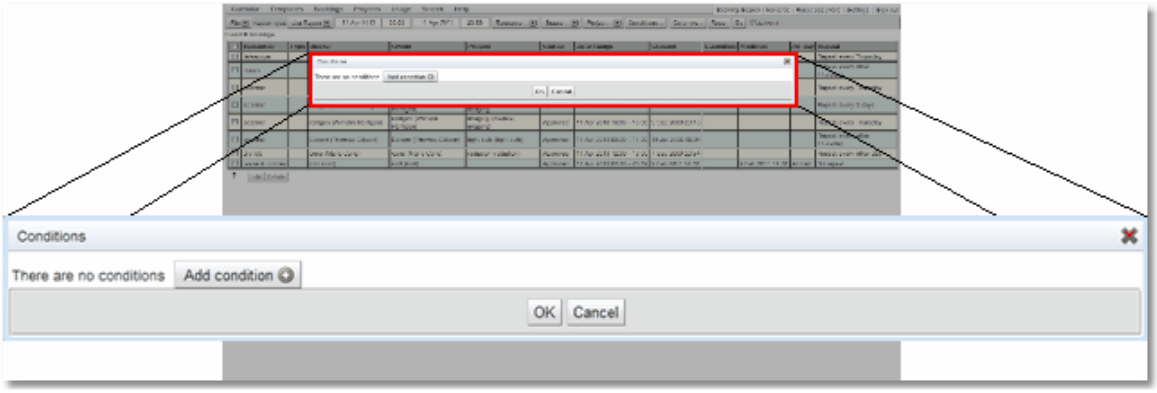
Approved:This is the normal *status* used for *projects* that are approved for making *bookings*[446].

Denied: a *project* request is turned down, then its *status* may be set to **Denied**.

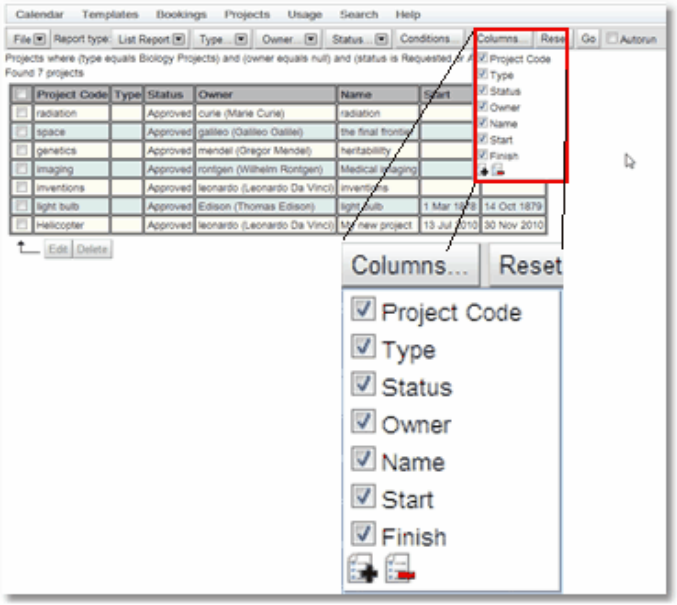
Unbookable:Some special *projects* may have a status of **Unbookable**. As the name suggests, these projects are not available for *bookings*. Typically, this is used only for the *project* that's used as a template for newly created *projects*.

Terminated: This is the *status* used for *projects* that are no longer active.

Additional *conditions*[447] for your search can also be set up. For more information on how to set up *conditions* read the information in the Search[108] page.



© 2010-2015 Exprodo Software



And finally change the columns displayed in the output.

Reset: Resets the search *conditions* to the defaults set up on starting the search.

Go: Runs the search with the current *conditions*

Autorun: If this button is ticked then searches will run as soon as any *conditions* change. If a number of the *conditions* of the search will be changed then it is more efficient if this is not ticked as each change will start a search and the appropriate transfer of data from the server.

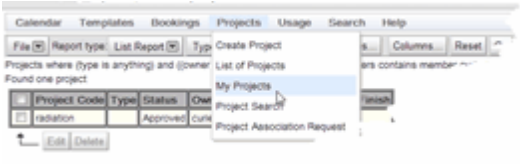
For a more complete explanation on how to edit records in a *project* search view read the How to Edit Multiple Items At Once[128] section of the Data Explorer[127] chapter. The only difference is in the booking search views as well as **Edit** and **Delete** there are also options to **Approve**, **Deny** or **Terminate** multiple *projects*.

© 2010-2015 Exprodo Software

3.5.3.2 My Projects

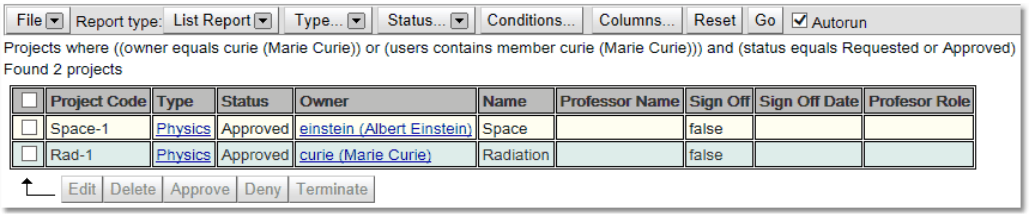
The **My Projects** search page does a search for *projects*[448] that the user have created or is a member off.

By default, the **My Projects** page appears on the menu here:



However, the administrator may have configured **Calpendo** so that the menu is different.

This is what the **My Projects** page looks like once a search has been done:


File
Report type: List Report
Type...
Status...
Conditions...
Columns...
Reset
Go
Autorun
Projects where ((owner equals curie (Marie Curie)) or (users contains member curie (Marie Curie))) and (status equals Requested or Approved)
Found 2 projects
Project Code | Type | Status | Owner | Name | Professor Name | Sign Off | Sign Off Date | Profesor Role
Space-1 | Physics | Approved | einstein (Albert Einstein) | Space | | false | |
Rad-1 | Physics | Approved | curie (Marie Curie) | Radiation | | false | |
Edit
Delete
Approve
Deny
Terminate

The **My Projects** search is run with the following default parameters:

Projects where ((**owner** equals **current_user**) or (**users** contains member **current_user**))) and (*status*[448] equals **Requested** or **Approved**)

Because the **owner** parameter is set specifically to **current_user** there is no **Owner** button available but the other buttons can be used to change the parameters of the search.

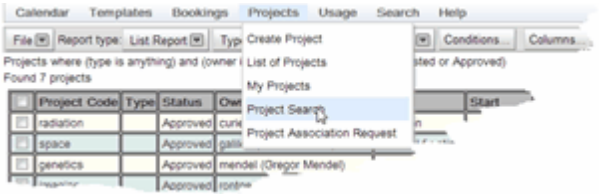
To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search[108] chapter.

© 2010-2015 Exprodo Software

3.5.3.3 Project Search

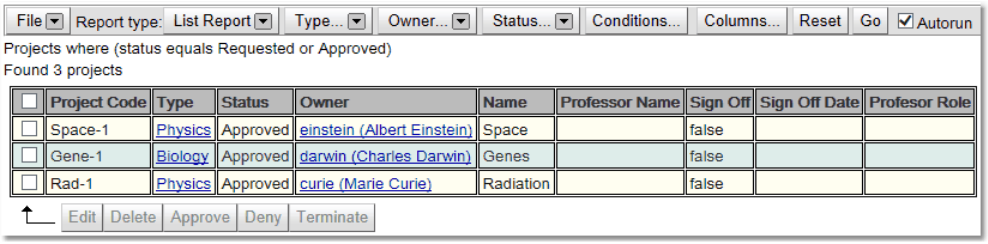
The **Project Search** page provides a convenient way to search for *projects*[448]. By default it returns all the **Requested** or **Approved** *projects*.

By default, the **Project Search** page appears on the menu here:



However, the administrator may have configured **Calpendo** so that the menu is different.

This is what the **Project Search** page looks like once a search has been done:



Project Code	Type	Status	Owner	Name	Professor Name	Sign Off	Sign Off Date	Profesor Role
Space-1	Physics	Approved	einstein (Albert Einstein)	Space		false		
Gene-1	Biology	Approved	darwin (Charles Darwin)	Genes		false		
Rad-1	Physics	Approved	curie (Marie Curie)	Radiation		false		

The **Projects Search** is run with the following default parameters:

Projects where (*status*[448] equals **Requested** or **Approved**)

All the possible buttons are available to change parameters of the search.

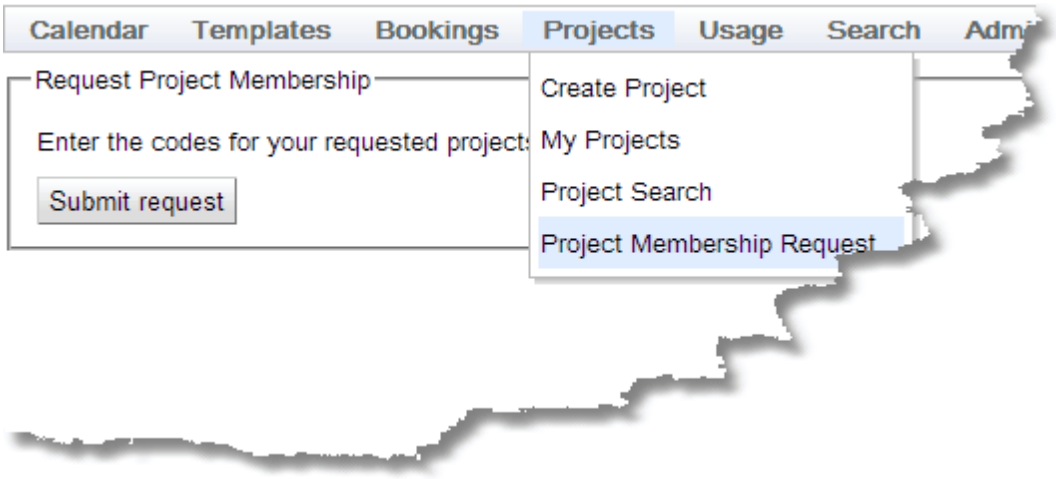
To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search[108] chapter.

© 2010-2015 Exprodo Software

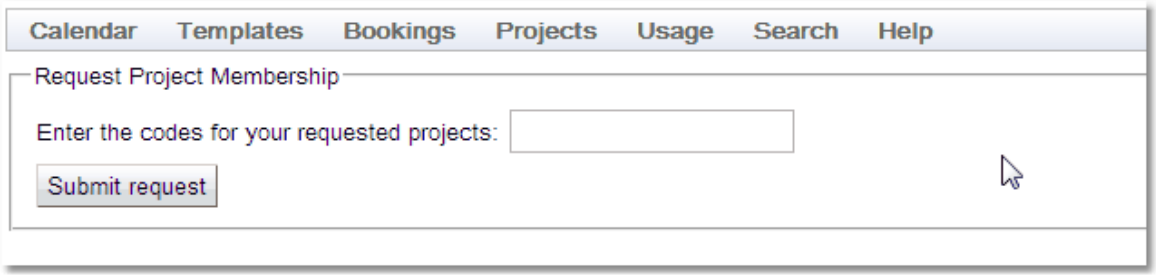
3.5.4 Project Membership Request

If a user would like to be associated with a *project*[448] but they don't have sufficient *permission*[448] to edit the *project* and add themselves, then they can ask the administrator or they can make a request using the **Project Membership Request** page.

The default place on the menu for this page is at **Projects-->Project Membership Request**. However, **Calpendo** may have been configured so that this page is called something else, or is not shown at all.



Enter the *code*[448](s) for the *project*(s) for which a joining request is needed, and press the **Submit Request** button. This will notify the administrator who will see whatever text has been entered. There is no automated response to this. Once the request has been submitted, wait for your administrator to respond to it.



© 2010-2015 Exprodo Software

3.6 Resource Usage

While **Calpendo** is primarily about *bookings*[446], it can also record the actual *usage*[449] of a *resource*[449]. This is done with a page that essentially offers a start and a stop button, and captures information about what the *resource usage* was for and whether it went well or not. Typically, this would be done with a computer that is next to the equipment whose *resource usage* is being recorded.

The administrator may have configured *resource usage* recording only for some *resources*, or for none of them. If *resource usage* recording is disabled for all *resources*, then the menu options relating to *resource usage* are not shown.

3.6.1 Resource Usage Session Recorder

The *Resource Usage Session Recorder*[449] provides the means for recording *resource usage*[449], and is split into three sub-pages:

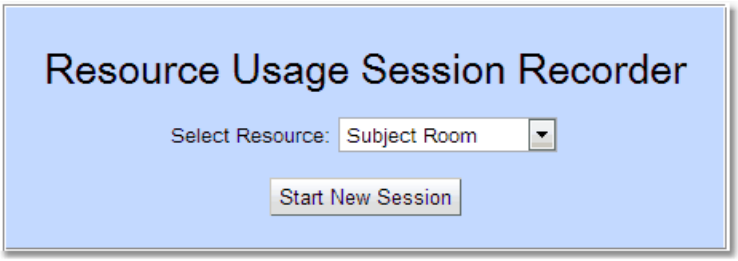
- The **Resource Selector** page that allows selection of the *resource*[449] to enter *usage* for. Once a *resource* is selected, either the pre-usage page or the during-usage page are open, depending on whether there is currently ongoing *resource usage* activity.
- The **Pre-Usage** page, that gathers information about what the *resource* is going to be used for and lets the user indicate that *resource usage* has actually started.
- The **During Usage** page that shows limited information about what the *resource* is being used for, when it started and how long it's been going for. It also provides a **Stop Session** button to indicate that *resource usage* has stopped.

A normal user can record *usage* for *resources* that do not need *projects*[448] associated with them or for *resources* where they are a member of the *project* that is using the *resource*.
An administrator can record *usage* for any *resource* and *project* combination.
This means that if the *resource usage* information is being recoded by operators they either need to be a member of all *projects* that use the *resource* they are operating or be logged in as an administrator.

Resource Selector Page

This page allows selection of which *resource* to enter *usage* for, or to view or terminate the current *resource usage* of.

If there are no current *resource usages* being collected then the following form will come up to choose the *resource* and start a new session and then move to the **Pre-Usage** page.



If there is a current *resource usage* being collected then the **During Usage** page will be opened.

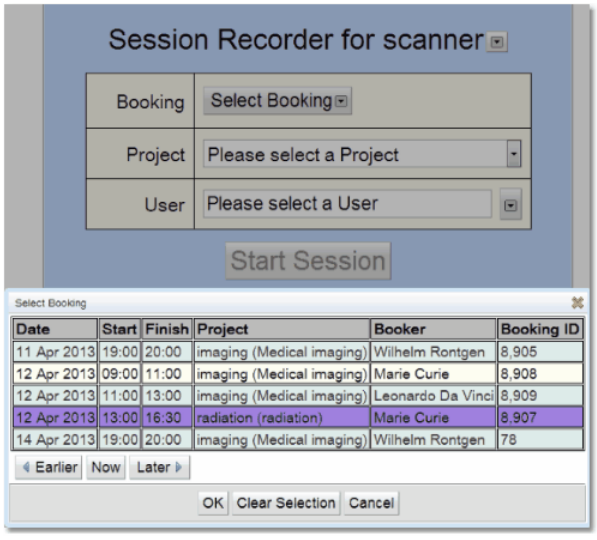
© 2010-2015 Exprodo Software

Pre-Usage Page

This page this lets you enter the following:

Setting	Description
Booking 446	Select a *booking* if a *booking* is to be associated with this *resource usage*. There may be no *booking* though, so this can be left unselected.
Project	This is only offered as an option if the *resource* has been configured to require a *project* for its *bookings*.
User	Specifies the user that is using the *resource*. A standard user may only select themselves. Administrators may select anyone.
Other	More information can be collected at this point, if required. This requires that the **Resource Usage** definition be modified by the administrator to add whatever additional information is required. For example, a user may want to capture information such as patient name, DOB and hospital number, whether this is a patient or healthy volunteer and any other required data. For each configured item, a user can specify whether **Calpendo** will store it in its database or not. (In a future version, it will be able to send data to another system via http/https, and this will include those things **Calpendo** doesn't store in its own database)

First select the *booking* to record *resource usage* for. When selecting a *booking*, the pop-up **Booking Selector** will display the *bookings* the user has *permission* 448 to access for the current *resource* around the current date and time. This should make it easy to select the right *booking*.


Session Recorder for scanner
Booking
Select Booking
Project
Please select a Project
User
Please select a User
Start Session
Select Booking
Date Start Finish Project Booker Booking ID
11 Apr 2013 19:00 20:00 imaging (Medical imaging) Wilhelm Rontgen 8,905
12 Apr 2013 09:00 11:00 imaging (Medical imaging) Marie Curie 8,908
12 Apr 2013 11:00 13:00 imaging (Medical imaging) Leonardo Da Vinci 8,909
12 Apr 2013 13:00 16:30 radiation (radiation) Marie Curie 8,907
14 Apr 2013 19:00 20:00 imaging (Medical imaging) Wilhelm Rontgen 78
Earlier Now Later
OK Clear Selection Cancel

If a *project* is requested, then one must be selected. Note that if a *booking* is selected, then the *project* will be automatically selected. Also select the user that is using the *resource*, here again *Permissions* may apply. Once the user, and possibly the *project*, is selected, then press the **Start Session** button when the *resource usage* begins. At this point, **Calpendo** records the time and other information entered, and displays the **During Usage** page.

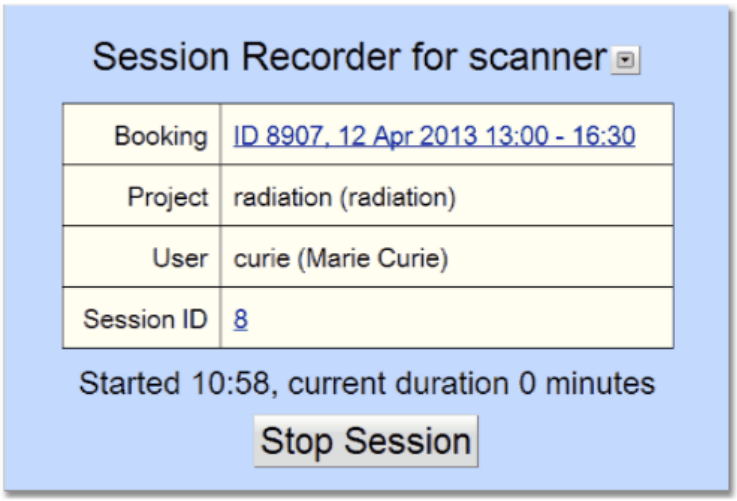
© 2010-2015 Exprodo Software

During Usage Page

This page shows the following information in a read-only format:

- The *resource*, together with a button to allow the *resource* to be changed, by going back to the **Resource Selector** page.
- The *project* (if applicable)
- The user
- The session ID that has been allocated. This displays as a hyper link that will display detailed information about the *resource usage* that has been recorded so far.
- The time that the *resource usage* started
- How long *resource usage* has been ongoing

There is also a **Stop Session** button, which should be pressed when *resource usage* ceases.



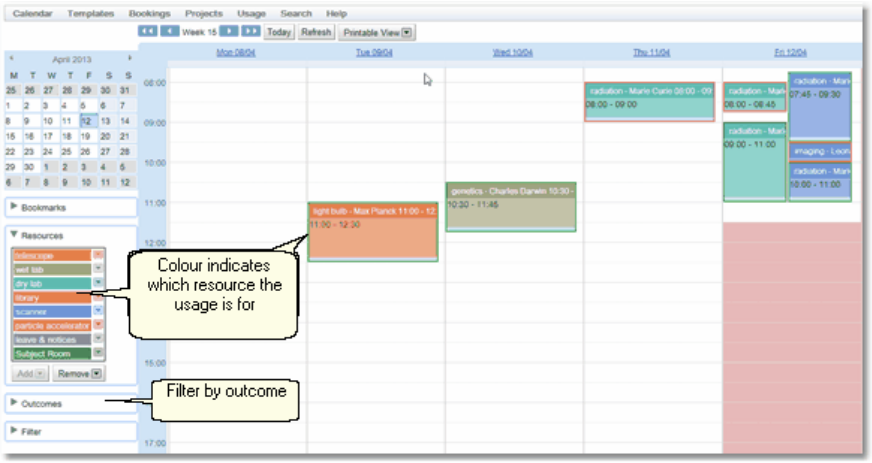
At this point, a pop-up will appear asking whether the session was successful. If the answer is no, then it asks for the outcome to be selected from a list of options that the administrator will have configured. The user may also provide a comment.

After completing the pop-up, the view will return to the **Pre-Usage** page.

© 2010-2015 Exprodo Software

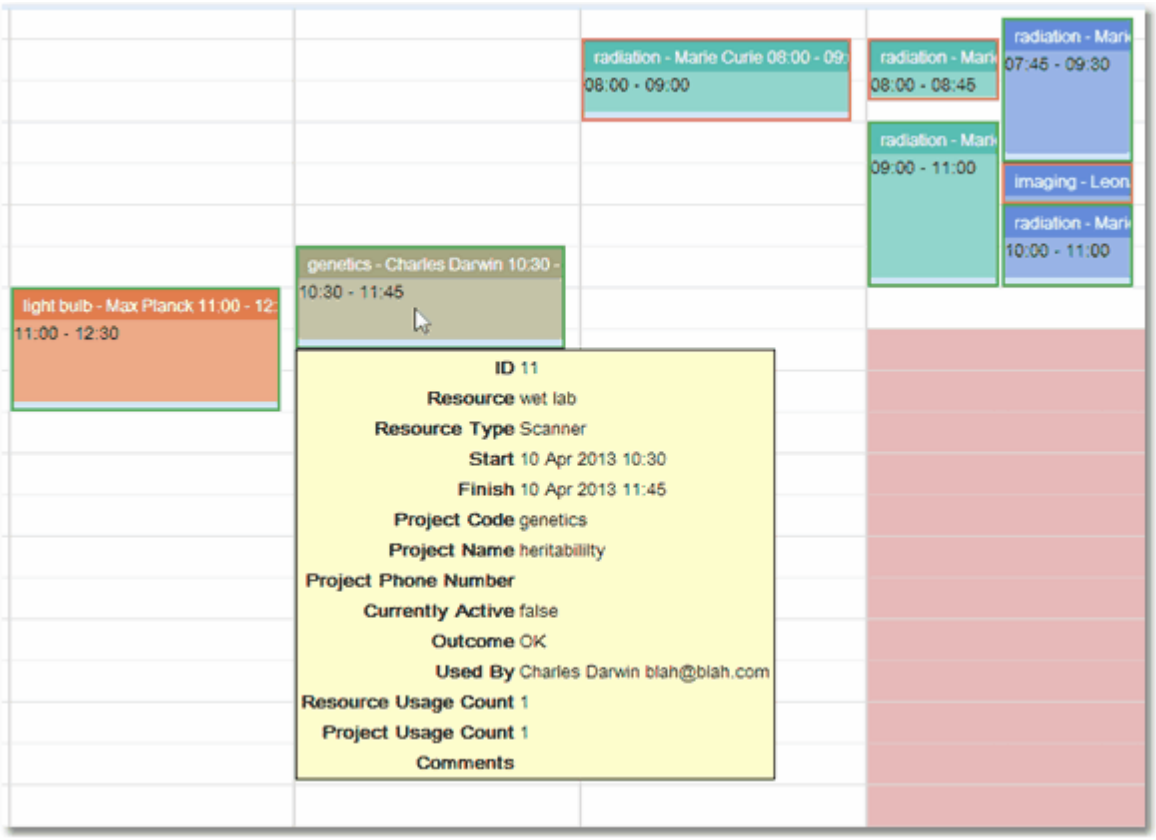
3.6.2 Resource Usage Calendar

The *Resource Usage Calendar*[449] shows a calendar much like the *Bookings Calendar*[446], except that everything is displayed in the past. That's because *resource usage*[449] that hasn't happened yet cannot be recorded. The calendar shows *usage* for any *resource* and gives the ability to filter that *usage* as required. If the user has suitable *permission*[448], then they can also create and modify *resource usage* from the *calendar*. This may be necessary if something prevented *usage* being recorded as a *resource* was being used.



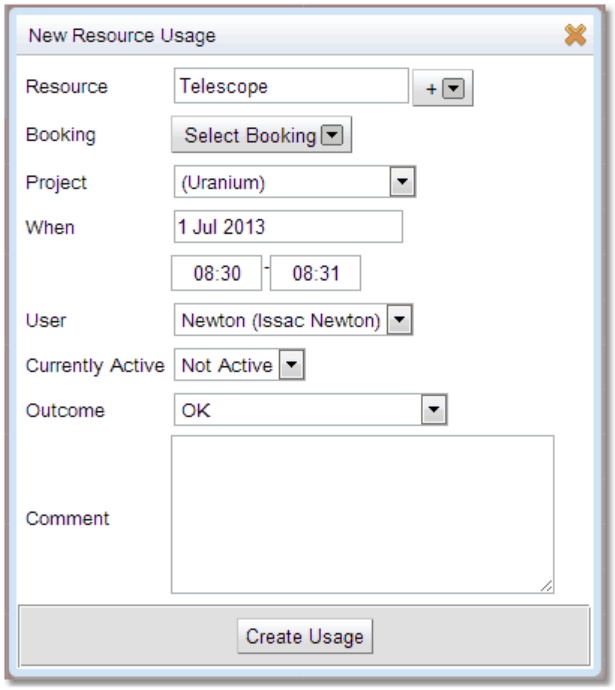
Filter by outcome allows the *resource usage's* shown to be filtered by the outcome of the *resource usage*. These outcomes will have been configured by the administrator.

By moving the cursor over a usage the *resource usage* information can be viewed.

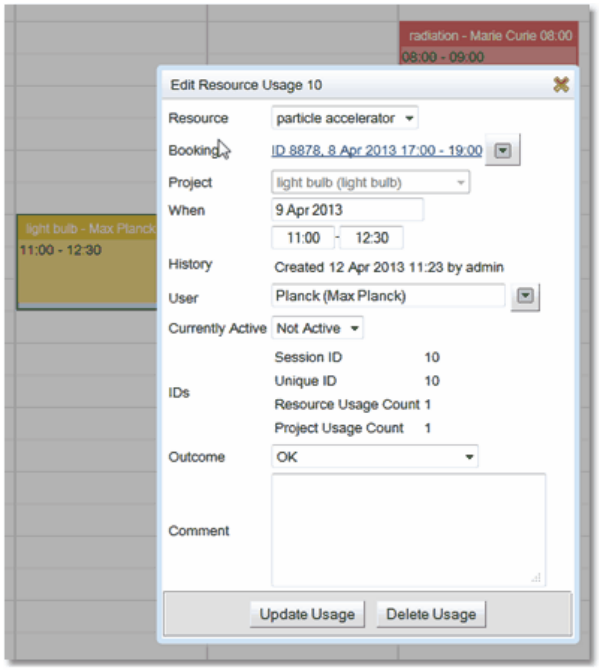


© 2010-2015 Exprodo Software

The user may need to create a *resource usage* record after the fact due to problems at the time, to do this select the time area for the resource usage and the resource usage create pop up will appear Fill in the *Resource*, *Booking*[446], *Project*[448] and **User** information. Once the **Outcome** is chosen the **Create Usage** button becomes active and can be pressed to create the *resource usage* information.



If the user has the *permissions* they can edit a resource usage's information. Click on the *resource usage* to be edited and choose edit from the pop up menu



Once the required edits are completed, press **Update Usage** to make the changes.

© 2010-2015 Exprodo Software

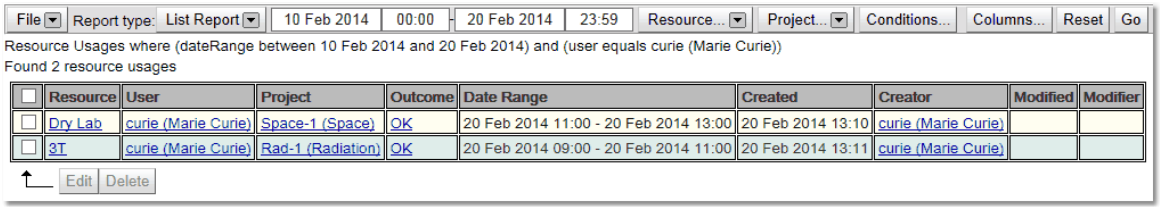
3.6.3 Searching For Resource Usage

Calpendo has multiple ways of finding out information about *resource usage*[449].

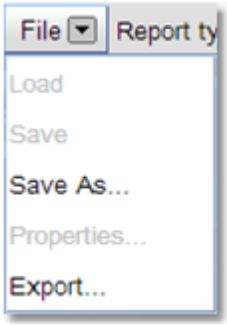
3.6.3.1 Using the Resource Usage Searches

All the *usage*[449] search pages work in a similar way. This chapter explains the basic mechanisms for all the *resource usage* searches. To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search[108] chapter.

When entered any of the *resource usage* search pages you will automatically start a search of the appropriate type with a date range of today. This is what a search will look like when it is completed.

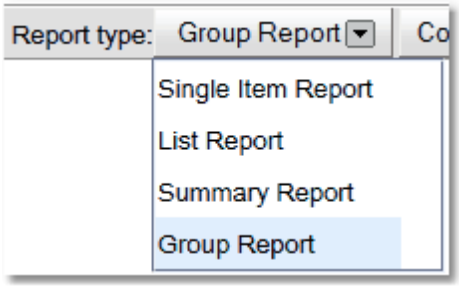


The following section goes through the options available after the search has completed, enabling the search parameters to be changed, saved, or the data exported, as well as change the information viewed by the search. Not all options will be available for every *resource usage* search type, but that will be outlined in the specific search types chapter. For more information on any of the features shown read the Search[108] chapter.



Save As: Saves the report for later use.

Export: Export the information found in the search to a file.



This button determines what type of report will be seen:

Single Item Report[449] Displays a single record that matches the search.

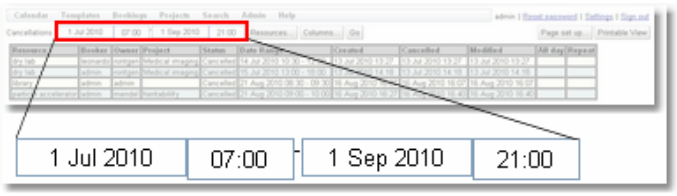
List Report[447]: Lists each record found that matches the search one under the other.

Summary Report[449]: Allows a summary of the information found.

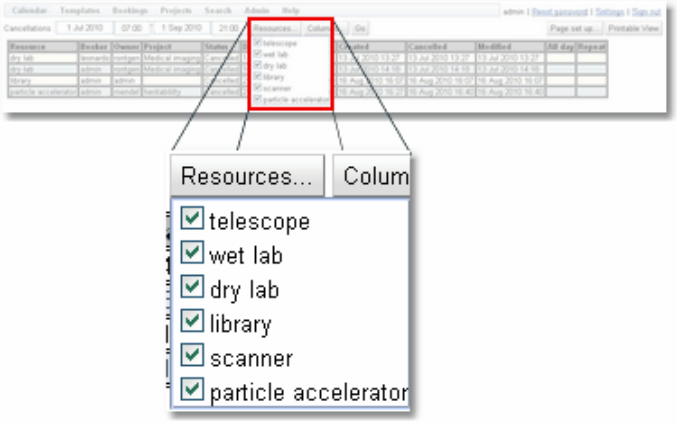
Group Report[447]: Allows the report to be shown by groupings with a count of the number of records that fit each group.

© 2010-2015 Exprodo Software

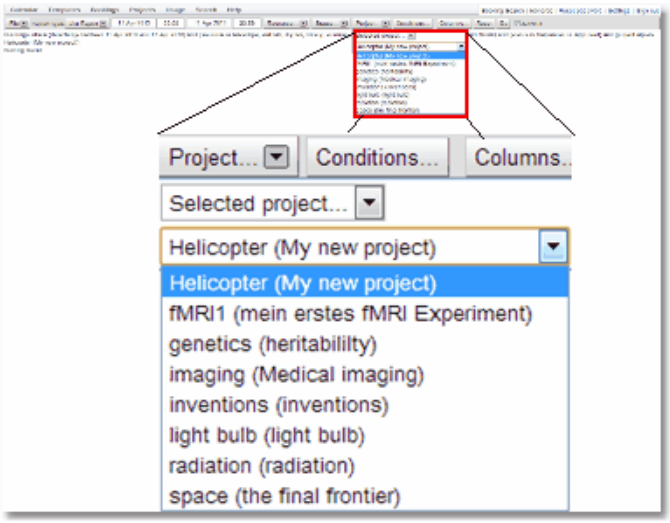
To change the search parameters



First select an appropriate range of dates for the new search.

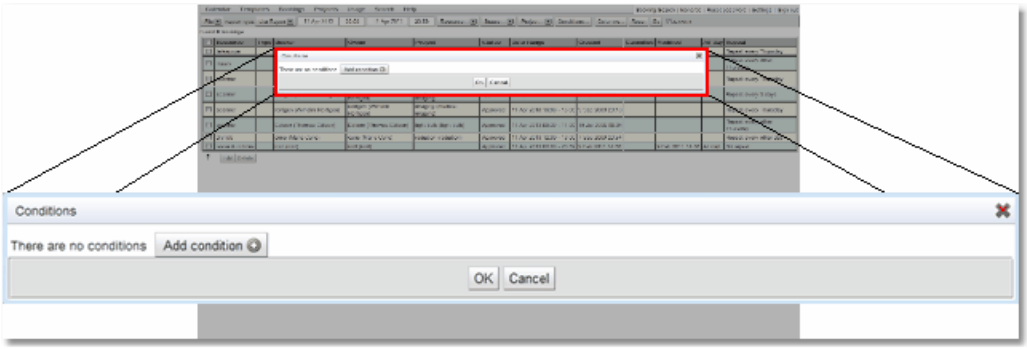


Then select the *resources*[449] required.

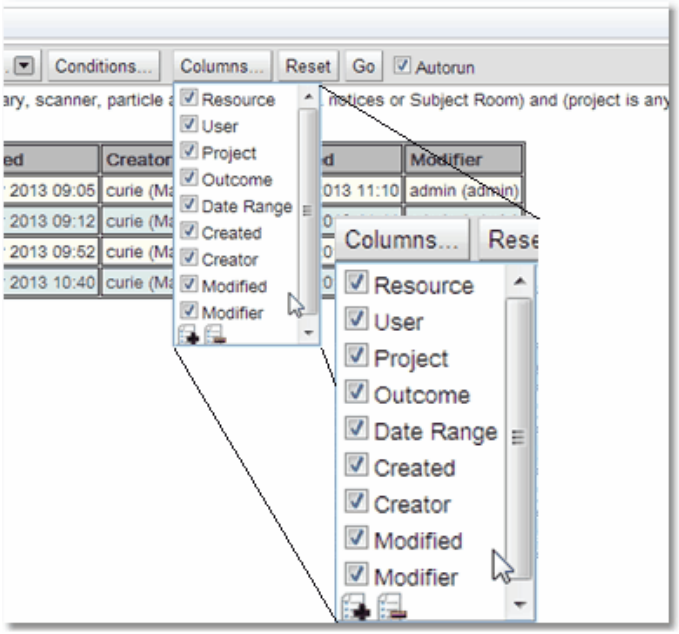


Then select the *project*[448] required.

Additional *conditions*[447] for the search may also be set up. For more information on how to set up *conditions* read the information in the Search[108] page.



© 2010-2015 Exprodo Software



Finally change the columns displayed in the output.

Reset: Resets the search *conditions* to the defaults set up on starting the search.

Go: Runs the search with the current *conditions*

Autorun: If this button is ticked then searches will run as soon as any *conditions* change. If a number of the *conditions* of the search will be changed then it is more efficient if this is not ticked as each change will start a search and the appropriate transfer of data from the server.

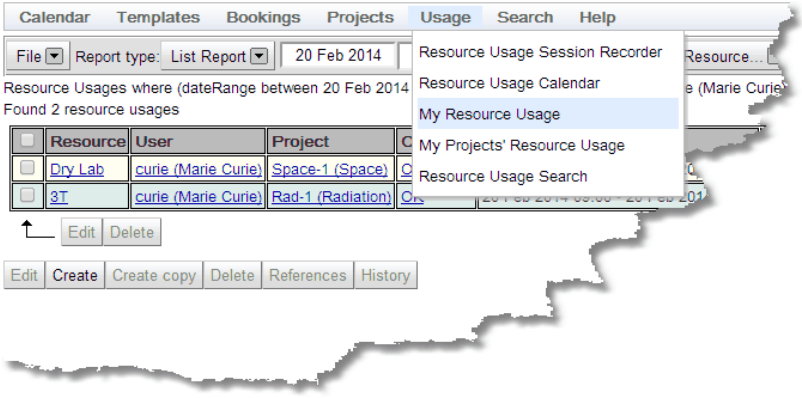
For a more complete explanation on how to edit records in a *usage* search view read the How to Edit Multiple Items At Once[128] section of the Data Explorer[127] chapter.

© 2010-2015 Exprodo Software

3.6.3.2 My Resource Usage

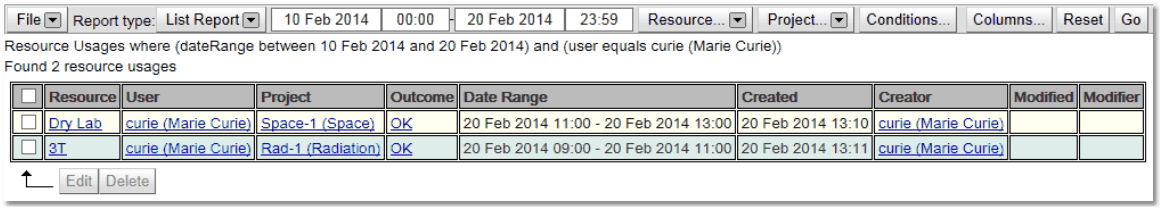
The **My Resource Usage** search page searches for *resource usage*[449], restricting any output to *resource usage* used by the current user.

By default, the **My Resource Usage** page appears on the menu here:



However, the administrator may have configured **Calpendo** so that the menu is different.

This is what the **My Resource Usage** page looks like once a search has been done:



The **My Resource Usage** search is run with the following default parameters:

Resource Usages where (**dateRange** between **start_date** and **end_date**) and (**user** equals **current_user**)

All the possible buttons are available to change parameters of the search.

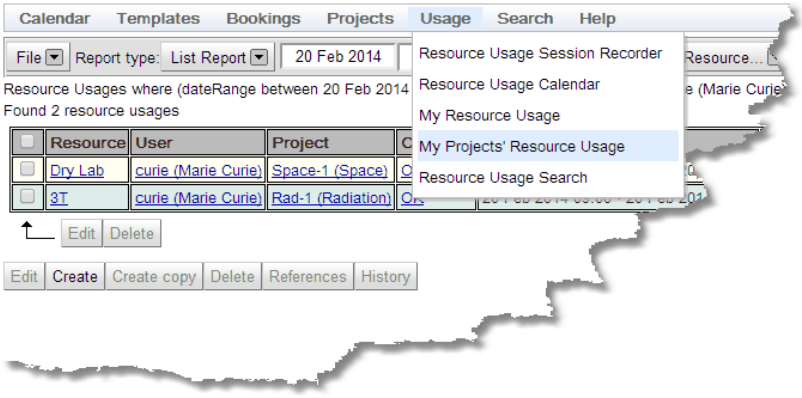
To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search[108] chapter.

© 2010-2015 Exprodo Software

3.6.3.3 My Projects' Resource Usage

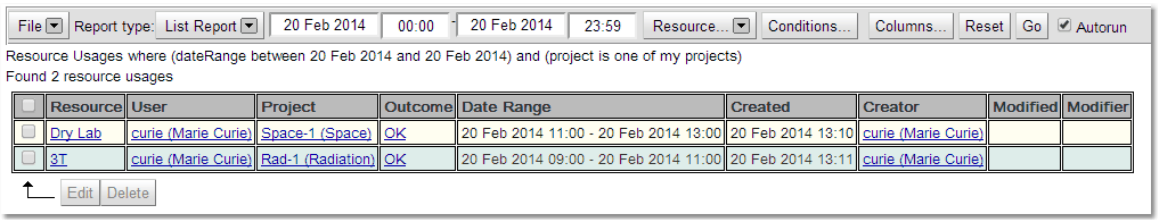
The **My Projects' Resource Usage** search page lets you search for *resource usage*[449] restricted output to *resource usage* used by the current user's *projects*[448].

By default, the **My Projects' Resource Usage** page appears on the menu here:



However, your administrator may have configured **Calpendo** so that the menu is different.

This is what the **My Projects'Resource Usage** page looks like once a search has been done:



The **My Projects' Resource Usage** search is run with the following default parameters:

Resource Usages where (**dateRange** between **start_date** and ***end_date***) and (*project* is **one of my projects**)

Because the *project* parameters is set specifically to **one of my projects** there is no **Project** button available but the other buttons can be used to change the parameters of the search.

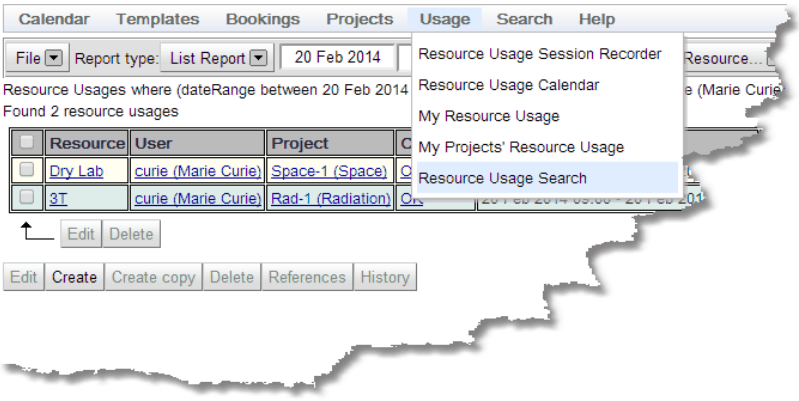
To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search[108] chapter.

© 2010-2015 Exprodo Software

3.6.3.4 Resource Usage Search

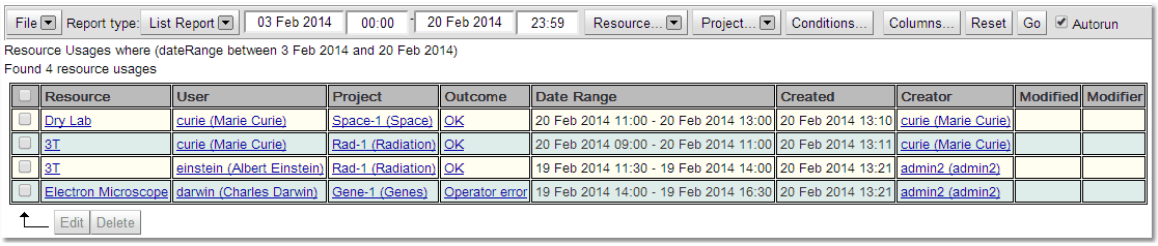
The **Resource Usage Search** page does a search through the *resource usage*[449] without any restrictions (apart from whatever *Permissions*[448] have been configured).

By default, the **Resource Usage Search** page appears on the menu here:



However, the administrator may have configured **Calpendo** so that the menu is different.

This is what the **Resource Usage Search** page looks like once a search has been done:



The **Resource Usage Search** is run with the following default parameters:

Resource Usages where (**dateRange** between **start_date** and **end_date**)

All the possible buttons are available to change parameters of the search.

To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search[108] chapter.

© 2010-2015 Exprodo Software

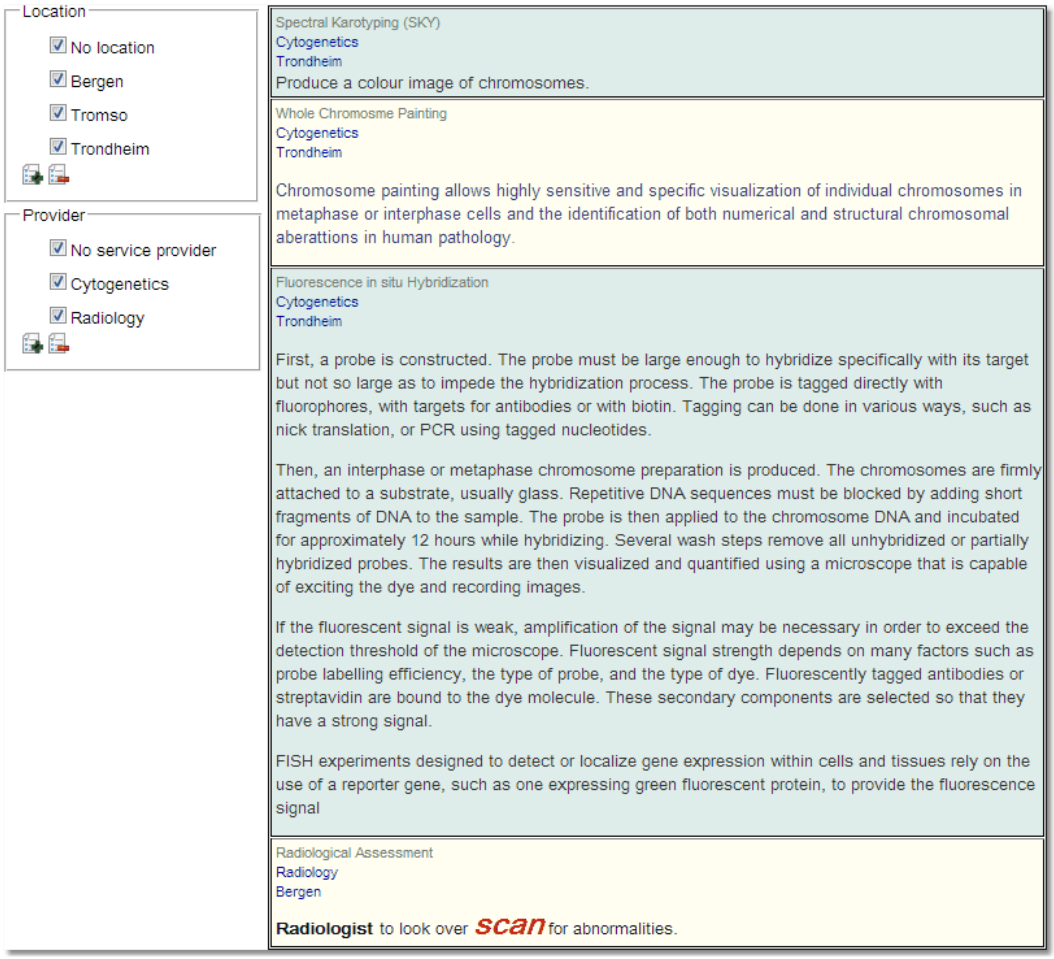
3.7 Services

While **Calpendo** is primarily about *bookings*[446], it can also be used to book services. A **Service** is additional functionality that a user can buy as part of their *project*[448] work. These can be such things as blood tests, checking scans, Chromosome painting etc.

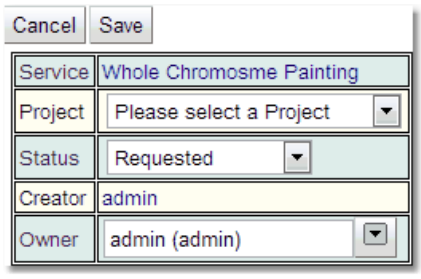
The administrator will have configured which *services*[449] are available for a user to order, created the order forms and set up the pricing structure for the cost of those *services*.

3.7.1 Available Services

This page lists all the *services*[449] that are available to be purchased at the facility. These can be filtered by location and provider. To order a *service* hover over the right hand top corner of the *service* and the order link will appear. Click on the link and the order process will start.


Location
No location
Bergen
Tromso
Trondheim
Provider
No service provider
Cytogenetics
Radiology
Spectral Karotyping (SKY)
Cytogenetics
Trondheim
Produce a colour image of chromosomes.
Whole Chromosme Painting
Cytogenetics
Trondheim
Chromosome painting allows highly sensitive and specific visualization of individual chromosomes in metaphase or interphase cells and the identification of both numerical and structural chromosomal aberattions in human pathology.
Fluorescence in situ Hybridization
Cytogenetics
Trondheim
First, a probe is constructed. The probe must be large enough to hybridize specifically with its target but not so large as to impede the hybridization process. The probe is tagged directly with fluorophores, with targets for antibodies or with biotin. Tagging can be done in various ways, such as nick translation, or PCR using tagged nucleotides.
Then, an interphase or metaphase chromosome preparation is produced. The chromosomes are firmly attached to a substrate, usually glass. Repetitive DNA sequences must be blocked by adding short fragments of DNA to the sample. The probe is then applied to the chromosome DNA and incubated for approximately 12 hours while hybridizing. Several wash steps remove all unhybridized or partially hybridized probes. The results are then visualized and quantified using a microscope that is capable of exciting the dye and recording images.
If the fluorescent signal is weak, amplification of the signal may be necessary in order to exceed the detection threshold of the microscope. Fluorescent signal strength depends on many factors such as probe labelling efficiency, the type of probe, and the type of dye. Fluorescently tagged antibodies or streptavidin are bound to the dye molecule. These secondary components are selected so that they have a strong signal.
FISH experiments designed to detect or localize gene expression within cells and tissues rely on the use of a reporter gene, such as one expressing green fluorescent protein, to provide the fluorescence signal
Radiological Assessment
Radiology
Bergen
Radiologist to look over scan for abnormalities.

Once the order option is clicked the appropriate *Service Order*[449] form will appear.



© 2010-2015 Exprodo Software

Fill in the order form and **Save** for the *service order* to be created. Above is the basic order form this can be changed by the administrator by adding new *properties*[448] either to the *Service Order* *Bisket Type*[446] or creating a new *Biskit Type* that derives from *Service Order* and includes any required additional *properties*.

3.7.2 Searching For Orders

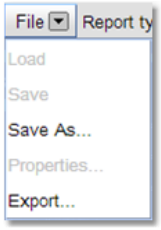
Calpendo has multiple ways of finding out information about the *service orders*[449] that have been created..

3.7.2.1 Using The Order Searches

All the order search pages work in a similar way. This chapter explains the basic mechanisms for all the order searches. To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search[108] chapter.

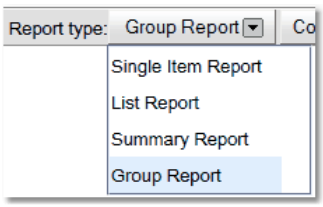
When entered any of the order search pages you will automatically start a search of the appropriate type. This is what a search will look like when it is completed.

The following section goes through the options available after the search has completed, enabling the search parameters to be changed, saved, or the data exported, as well as change the information viewed by the search. Not all options will be available for every order search type, but that will be outlined in the specific search types chapter. For more information on any of the features shown read the Search[108] chapter.



Save As: Saves the report for later use.

Export: Export the information found in the search to a file.



This button determines what type of report will be seen:

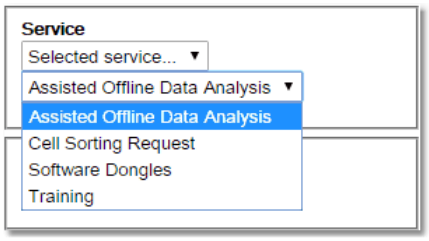
List Report[447]: Lists each record found that matches the search one under the other.

Summary Report[449]: Allows a summary of the information found.

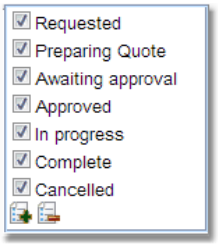
Group Report[447]: Allows the report to be shown by groupings with a count of the number of records that fit each group.

© 2010-2015 Exprodo Software

To change the search parameters

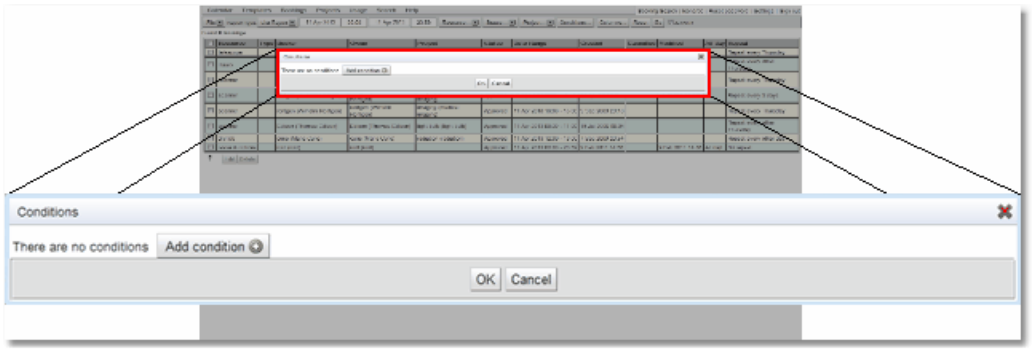


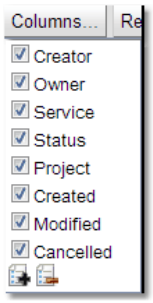
There are a number of options that can be used to filter the services. They all provide a drop down menu so that a choice between any, none or a particular item can be made. The filters are: **Owner**, **Creator**, **Service**, **Provider**, **Location**.



The final filter is **Status**. This filter allows you to use tick boxes to choose which status('s) to filter on.

Additional conditions for the search may also be set up. For more information on how to set up conditions read the information in the Search 108 page.





Finally change the columns displayed in the output.

© 2010-2015 Exprodo Software

Reset: Resets the search conditions to the defaults set up on starting the search.

Go: Runs the search with the current conditions

Autorun: If this button is ticked then searches will run as soon as any conditions change. If a number of the conditions of the search will be changed then it is more efficient if this is not ticked as each change will start a search and the appropriate transfer of data from the server.

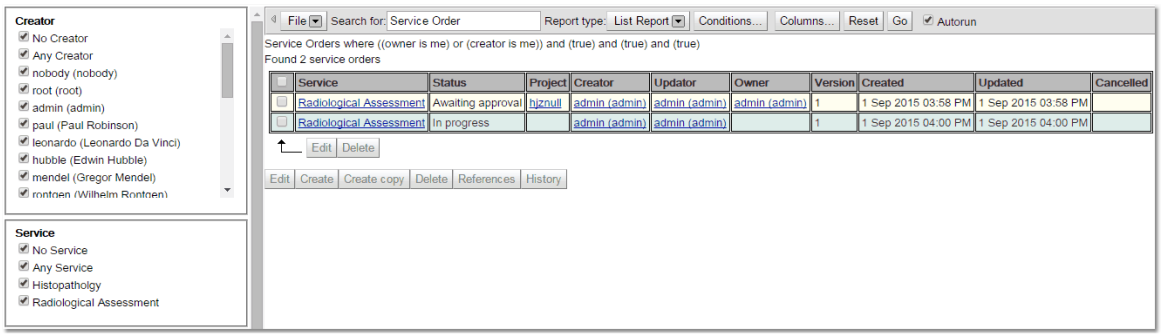
3.7.2.2 My Orders

The **My Orders** search page does a search for orders that the user has created.

By default, the **My Orders** page appears on the menu here:

However, the administrator may have configured **Calpendo** so that the menu is different.

This is what the **My Orders** page looks like once a search has been done:



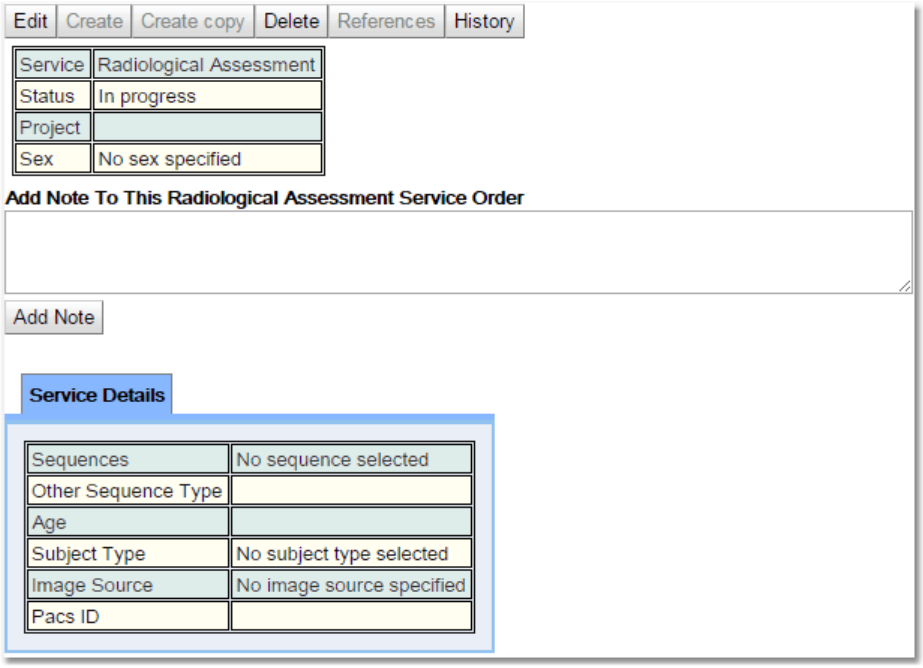
The my orders search is run with the following default parameters:

Service Orders where ((**owner** is **me**) or (**creator** is **me**))

Because the *owner* parameter is set to be **me** there is no **Owner** filter available, but the other filters found down the left of the page can be used to change the parameters of the search.

© 2010-2015 Exprodo Software

Once an order has been found, click on it for the full details to be shown. Once an order has been created if the administrator implemented **Notes** both the users dealing with the order and the initiator of the order can add **Notes**. To add a note, write in the box marked **Add Note To This .... Order** and then press the button. There is no need to be in edit mode for this to work.



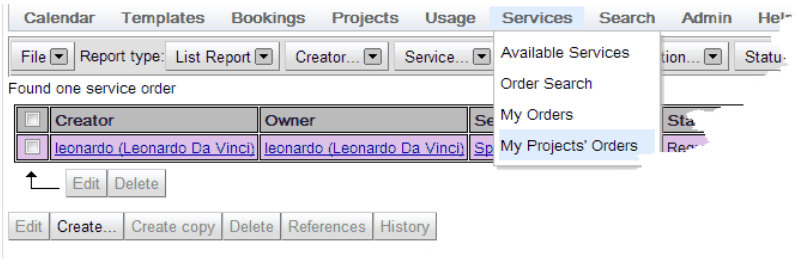
To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search 108 chapter.

© 2010-2015 Exprodo Software

3.7.2.3 My Projects' Orders

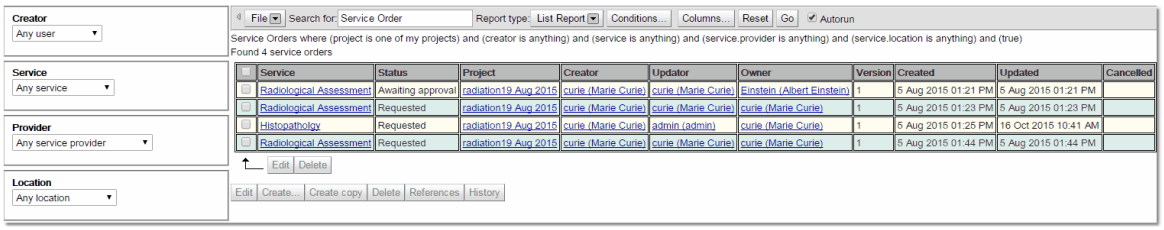
The **My Projects' Orders** search page does a search for orders that are associated with *projects* 448 that the user is also associated with.

By default, the *My Projects' Orders* page appears on the menu here:



However, the administrator may have configured **Calpendo** so that the menu is different.

This is what the **My Projects' Orders** page looks like once a search has been done:



The my orders search is run with the following default parameters:

Service Orders where (*project* is **one of my projects**) and (*creator* is **anything**) and (*service* is **anything**) and (*service.provider* is **anything**) and (*service.location* is **anything**)

Because the *project* option is **one of my projects** there is no **Owner** filter available but the other filters down the left of the page can be used to change the parameters of the search.

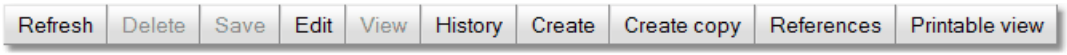
See the section on My Orders 93 to learn how to add comments to an order.

To learn more about the advanced features available with searches and how to select and edit data returned from a search then go to the Search 108 chapter.

© 2010-2015 Exprodo Software

3.8 Toolbar Button Standard Definitions

Calpendo uses toolbars in a number of places to provide options for editing and displaying information. This chapter will define the standard button operations. Below is an example of a toolbar with some options greyed out.



When a toolbar is being described, the text will refer to this chapter, and describe either those buttons not mentioned here or whose operation is significantly different from the norm.

Editing Buttons

Button	Description
Edit	The **Edit** button is greyed out if there is no item selected, or the user does not have *permission* 448 to edit the item. Pressing the button put the item into **Edit** mode. There will be a warning if another user is already editing the *Biskit,* and if the *Biskit* has the **Version Property** *meta-property* 448 set up, then the user will be warned if an older version is being edited,
Apply	The **Apply** button is active only while an item is in edit mode. Press it to save the item, and stay in edit mode.
Save/OK	The **Save/OK** button is active only while an item is in edit mode. Press it to save the item, and the item will revert back to a read-only view.
Cancel	The **Cancel** button is active only while an item is in edit mode. Press it to throw away any changes since the last **Apply** (if there is an **Apply** option on the menu) and move out of edit mode.
View	The **View** button is used to revert back to a view of the item's details, in read-only mode. If there is no item selected, or the item's details are already shown in read-only mode, then the button will be greyed out.
Delete	The **Delete** button is greyed out if there is no item selected, the selected item cannot be deleted due to *Permissions,* or it is being *referenced* 449 by something else. If there are many things in the database that reference the item, then it may be difficult to delete it, because the *references* must be removed first. Press to delete the selected items from the database.
Create	The **Create** button will create a new item of the selected type, but it will be greyed out if the user does not have *permission* to create items of that type. Once pressed, the item will go into edit mode. The item will not exist in the database until either **Save** or **Apply** are pressed.
Create Copy	The **Create Copy** button will create a copy of the selected type, but it will be greyed out if the user does not have *permission* to create items of that type. Once pressed, the item will go into edit mode. The item will not exist in the database until either **Save** or **Apply** are pressed.
Cut	The **Cut** button cuts the currently selected item, removing it from its current position and puts it in the edit buffer for later use, overwriting the current buffer contents.
Copy	The **Copy** button takes a copy of the currently selected item and puts it in the edit buffer for later use, overwriting the current buffer contents.
Paste	The **Paste** button pastes the current paste buffer contents to the currently selected position.

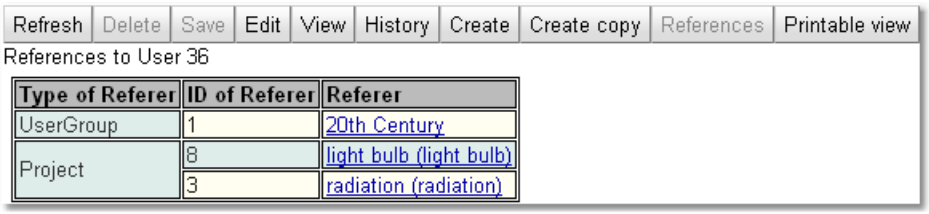
© 2010-2015 Exprodo Software

Viewing Buttons

Button	Description
Refresh	The **Refresh** button reloads the current page from the database.
Open All	The **Open All** button opens up the whole tree so that all the information can be seen.
Close All	The **Close All** button closes the tree information so only the top of each tree can be seen.
Printable View	The **Printable View** button will reformat the page to make it printer friendly, including removing the list of items and retaining only the selected item. Then use the browser's **Print** option to print the item's details. Click anywhere on the page to revert back to the normal view.
Columns...	The **Columns...** button allows the user to decide which *properties* 448 will be displayed. The default is always the **Name** *property*.

The References Button

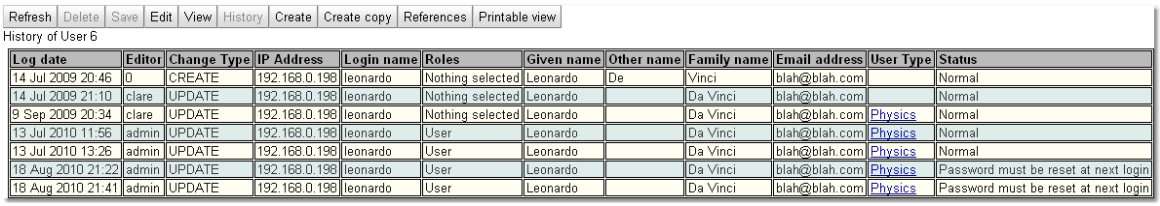
This button will show a list of everything that *references* the currently selected item. It will be greyed out if there is nothing referring to the item. Once pressed a list of the *references* will be viewed:



This shows a table with the *Biskit Type* 446 of the referrer, the unique database identifier of the referrer and its name. Either press the **References** button again, or press the **Back** button on the browser, to return to the original view.

The History Button

This will show a change *history* 447 of the selected item, like this:


Refresh
Delete
Save
Edit
View
History
Create
Create copy
References
Printable view
History of User 6
Log date
Editor
Change Type
IP Address
Login name
Roles
Given name
Other name
Family name
Email address
User Type
Status
14 Jul 2009 20:46 0 CREATE 192.168.0.198 leonardo Nothing selected Leonardo De Vinci blah@blah.com Normal
14 Jul 2009 21:10 clare UPDATE 192.168.0.198 leonardo Nothing selected Leonardo Da Vinci blah@blah.com Normal
9 Sep 2009 20:34 clare UPDATE 192.168.0.198 leonardo Nothing selected Leonardo Da Vinci blah@blah.com Physics Normal
13 Jul 2010 11:56 admin UPDATE 192.168.0.198 leonardo User Leonardo Da Vinci blah@blah.com Physics Normal
13 Jul 2010 13:26 admin UPDATE 192.168.0.198 leonardo User Leonardo Da Vinci blah@blah.com Physics Normal
18 Aug 2010 21:22 admin UPDATE 192.168.0.198 leonardo User Leonardo Da Vinci blah@blah.com Physics Password must be reset at next login
18 Aug 2010 21:41 admin UPDATE 192.168.0.198 leonardo User Leonardo Da Vinci blah@blah.com Physics Password must be reset at next login

This shows who has changed the item and when, and also shows some of the item's *properties* 448 as they changed over time. The "complex" *properties* are not shown. **Calpendo** does not record the history of *properties* that store lists of items, like lists of *projects* 448 and lists of *user groups* 450. The **History** button will be greyed out if there is no item selected or if the item's *history* is already being shown. To revert back to showing the item's details, press the **View** button or the **History** button.

© 2010-2015 Exprodo Software

3.9 Conditions

Conditions[447] are used in several places within **Calpendo**, most notably Searching[108], Automatic Emails[297] and Permissions[278] and Booking Rules[223]. Understanding *Conditions* is essential in order to be able to configure **Calpendo** properly because of their pervasive use. Their purpose is to allow you to specify what situations generate an email, or to which situations a Permission[278] or Booking Rule[223] applies. The chapter on Setting Search Conditions goes through basic *conditions* and how to use the **Conditions Panel**. This chapter will expand on how to use *conditions* for the more complex tasks required in Automatic Emails[297], and Permissions[278] and Booking Rules[223].

The Anatomy Of A Condition

A *Condition*[447] extracts one or two *properties*[448] from the context and checks whether a *relation*[449] holds true. If it does, the *Condition* is said to match. A *Condition* consists of the following items, in this order:

The **Condition Type** indicates the source of a *property*.
The *Property Path*[449] tells how to reach a *property* from that source.
The *Relation* provides an operator to compare the value of that *property* to another value.
The **Value** provides a value to do the comparison against.
The **Accuracy** specifies how accurate the comparison should be (used with dates).

The **Condition Type** and *Property Path* are always present, but the others may not be. For example, if the user hasn't selected a *Property Path*, then the *Relation*, **Value** and **Accuracy** do not show. If the context does not specify a *Biskit Type*[446], then when a *Condition* is first created, it will look like this:

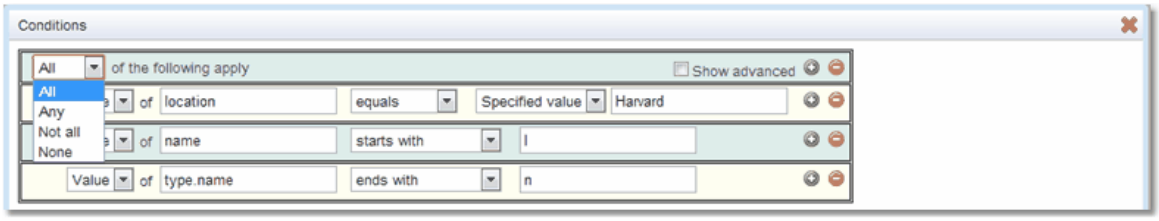
The above picture shows the default **Condition Type** of **Value** with a red border because **Value** cannot be used without a *Biskit Type* being available in the context. The *Property Path* also has a red border and shows there are no properties available for the same reason. Choosing a *Biskit Type* of anything other than **Any data type** changes the *Property Path* selector to show there are now *properties* that can be selected: :

The *Property Path* selector still shows in red because a *Property Path* is required, but hasn't been selected yet. Clicking where it says **Select a property** would now generate a drop-down menu with a list of all the *properties* that exist on the *Biskit Type*[446] selected in the current context.

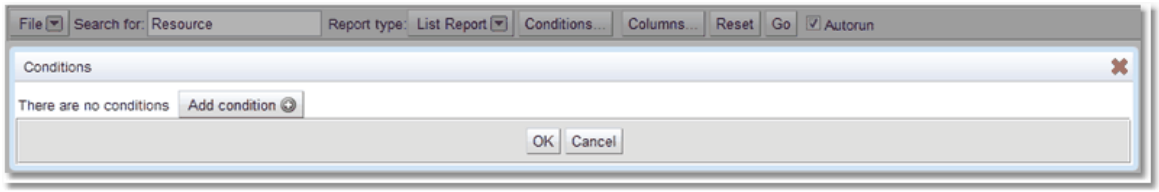
© 2010-2015 Exprodo Software

The Conditions Panel

Everywhere that *conditions* are used, they appear in a panel that looks something like this in a writeable context:



When there are no *conditions*, it looks like this in a writeable context:



When editing *Conditions*, add a new *Condition* by pressing the **Add Condition** button. At any time clicking on the **green +** button will add a new *condition* after the current one, clicking the **red -** button will delete the *condition* the button is associated with.

Whenever there are multiple *Conditions*, they will appear in the **Conditions Panel**, each in their own row.

The Conditions Context

A **Conditions Panel** is subject to the context within which it is used. There are two parts to the context that affect the way the **Conditions Panel** behaves: the *Biskit Type* 446 the *condition* is being placed upon, and whether used for an update so that there are two instances of the data available (the **Old Value** and the **New Value**) or whether there is just one instance available (the **Value**).

For an Automatic Email 297, the context *Biskit Type* is the type that triggers the email. If the email is triggered by any *Bisikit Type*, then it means the context has no *Bisikit Type*. There's a **New Value** available when the action that triggered the email is an **Update** and not when the action is **Create** or **Delete**.

For a Permission 278, it's a very similar situation. The context *Bisikit Type* is the *Bisikit Type* that the *Permission* 448 controls access to, and if the *Permission* applies to any *Bisikit Type*, then the context has no *Bisikit Type*. The **New Value** is available when the action being controlled is **Update** and not when the action is anything else.

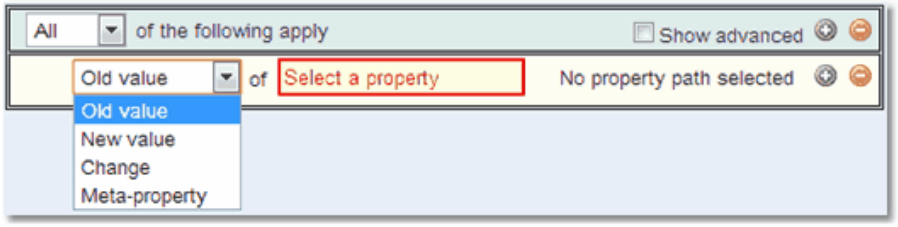
For a Search 108 the context *Bisikit Type* is the *Bisikit Type* that the **Search** is on, and the only **Condition Type** available is **Value**.

For Booking Rules 223, the context *Bisikit Type* is *Booking* 446. The **New Value** is available when the action being controlled is **Update** and not when the action is anything else.

© 2010-2015 Exprodo Software

The Conditions Type

When a *condition* is created or edited, the type has four possible values:



Condition Type: Old Value

A **Condition Type** of **Old Value** means the *condition* will be placed upon the value of a *property* 448 before any change has been made to it. For a *condition* within an Automatic Email 297 this means the value of the *property* in a **Create** or **Delete** action and the old value of the *property* for an **Update** action. For a Permission 278 this means the value of the *property* on an object somebody is trying to **Create** or **Delete**, and the original value they are trying to change from when somebody is trying to **Update** a *Biskit* 446. For a Booking Rule 223 this will be the value of the *property* on the **Booking** *bisikit type*.

A **Condition Type** of **Old Value** is only suitable when the *condition* context specifies a *bisikit type*.

When a **Condition Type** of **Old Value** is specified, the *property path* 449 will show all the *properties* of the type defined by the context's *bisikit type*, and any *properties* that can be reached from those *properties*. When the context *bisikit type* is set to a **Resource**, clicking where it says **Select a property** shows this:

A *Resource* 449 has a number of *properties*, as shown. One of them, **Location**, is itself an object that has *properties* of its own that can be selected. As soon as you select a *property* from the drop-down, the *Relation* 449 part of the *condition* will change from **No property path selected** to something appropriate for the *property* selected (described further below).

Condition Type: New Value

A **Condition Type** of **New Value** is only available when a *bisikit type* has been specified, and even then only in some contexts. For an Automatic Email 297, Permission 278 or Booking Rule 223, a **Condition Type** of **New Value** is only available for the **Update** action. Selecting **New Value** means that the *condition* will be placed on the value of the *property* after the update has occurred. The *property path* will display the same as for a **Condition Type** of **Old Value**.

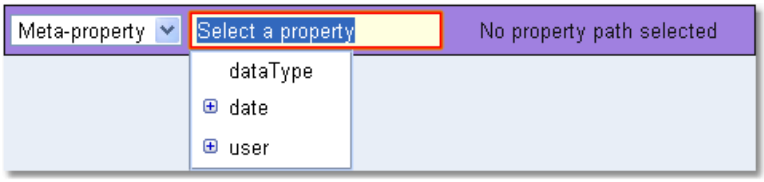
© 2010-2015 Exprodo Software

Condition Type: Change

A **Condition Type** of **Change** is available in the same contexts as **New Value** and the *property path* displays the same as for both **Old Value** and *New* **Value**. This **Condition Type** means that a restriction is placed upon the change of a *property*. For example,it can be restricted so that the *condition* matches when the selected *property* does not change, does change, increases or decreases. The increase and decrease restrictions are only available for *properties* for which such a concept makes sense. This is described further below.

Condition Type: Meta-Property

A **Condition Type** of **Meta-Property** is available in all contexts. When this type is selected, the *property path* displays the following:



A *Meta-Property* 448 lets you place restrictions on:

- the type of the data that was **Created**, **Updated** or **Deleted**
- the date and time that the action took place
- the user that caused the action

This is called a *Meta-Property* because it's not a *property* of whatever was created, updated or deleted, but rather, it's a *property* of the action itself.

The Property Path

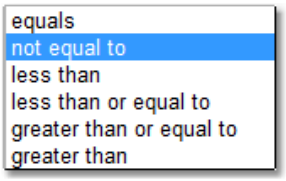
As described above, the *properties* shown in the *property path* depend upon the **Condition Type** selected. However, each *property* that is shown has it own way of being displayed, regardless of how it came to be there. For example, any date/time *property* that is shown will display with sub-properties.

A *Resource* 449 has a number of properties, as shown. One of them, **Location**, is itself an object that has *properties* of its own that can be selected. The *Relation*, **Value** and **Accuracy** part of the *condition* will not show unless a *property path* has been selected. Once a *property* is selected, the extra parts of the *condition* appear, and they reflect the type of the *property* selected and the choice of **Condition Type**.

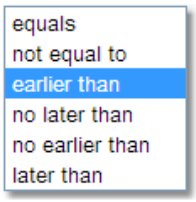
© 2010-2015 Exprodo Software

The Relation

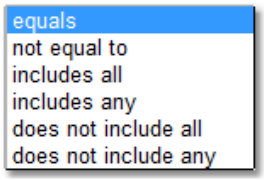
The *Relation* is only shown when there's a *bisikit type* in the context and a *property path* has been selected. The options displayed in the *Relation* drop-down menu differ depending on the type of *property* selected by the *property path*.



This is the standard choice of relations for an integer or floating point number property.

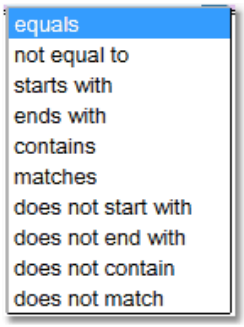


A date *property* displays with semantically identical options, but using different words to match the English you would expect.



Not all integer *properties* display using the above options. That's because some integer *properties* are defined as a bit set of values of named integers (see Bit Sets[368] for details). A good example of this is a *user's roles*[450]. There are up to 32 roles defined, and a user may be deemed to have any number of these roles. For such *properties* it doesn't make sense to compare two sets of roles using greater than or less than, and so the available *relations* are changed to reflect this.

A text *property* displays with options for comparison with the starting, ending or interior characters. Also available is pattern matching using the following:

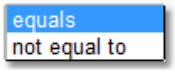


^	The beginning of the line.
$	The end of the line.
.	Any character.
*	Zero or more times.
+	One or more times.
?	Once or not at all.
X | Y	Either *X* or *Y*.
{n}	Exactly *n* times.
{n,m}	At least *n* times but not more than *m* times.
[a-z]	Characters a to z.
[^a-z]	Any character other than a to z

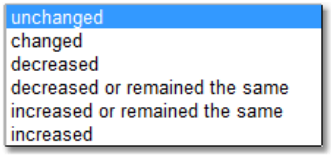
Search[108] conditions are not case sensitive, while non-search (Automatic Email[297], Permission[278] or Booking Rule[223]) conditions are case sensitive.

Depending on whether the pattern matching is going to be done in Java (Automatic Email[297], Permission[278] or Booking Rule[223]) or SQL (Search[108]) will depend on which additional matching features are available. It is recommended that users only use the options listed above.

© 2010-2015 Exprodo Software



All the remaining *property types* don't support any notion of an ordering that could be used to specify a **less than** *relation* etc. So for these, the relations available are.



All the above *relations* are used when the **Condition Type** is **Value**, **Updated Value** or **Meta-Property**. For a **Condition Type** of **Change**, the *relations* are different. In this case, a comparison between new and an old value are being done. The numeric options available are semantically identical to their original values, but to make it clear what is meant by **less than** in this situation, the wording changes to this.

All the remaining *property* types, including those that used the set of Mapped Integers[366] like *user roles* above, do not support any notion of comparing the new and the old value except for indicating whether there was a change.

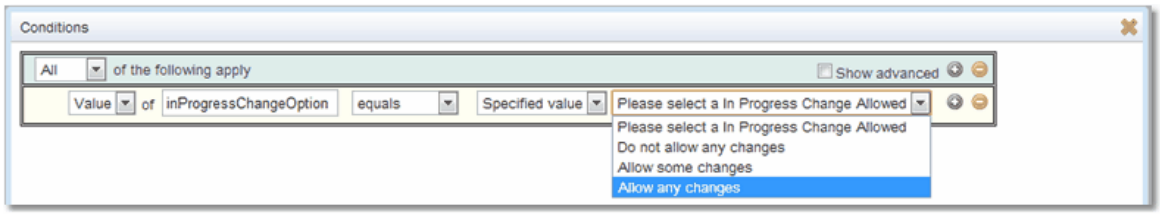
© 2010-2015 Exprodo Software

The Value

A value is not required when the **Condition Type** is **Change**. That's because the **Change** *relation* compares the new and the old values of the *property*, and so does not require a user-specified value to compare against.

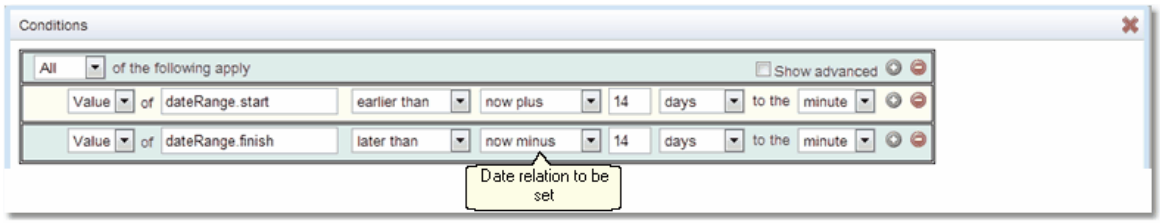
When a value is required, a drop-down menu is shown. Any *property* that is not a date or date-time, it shows this:

This gives a choice between a value that is **null** (a special value meaning there is no value) and **Specified value**. If **Specified value** is chosen then a further item will appear so that the value can be chosen. The item shown for the value entry will be appropriate to the type of the *property* indicated by the *Property Path*.

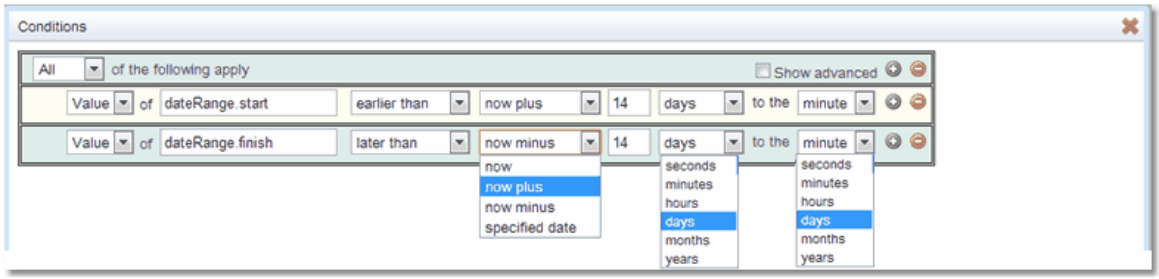


Date Properties And Accuracy

When dealing with dates there are some other options that are available. Whenever a *property* is asked for which stores a date two *condition* lines will appear, one for the start date and one for the finish date, the *relation* can then be set (see table earlier in the chapter for the full list available) :



Then set how accurate the date checking needs to be. There are four options against which the comparison can be made and then a level of accuracy to be used.



© 2010-2015 Exprodo Software

Date Accuracy	Description
now	The time specified but with a level of accuracy.
now plus	Within now plus a certain amount of time with a level of accuracy.
now minus	Within now minus a certain amount of time with a level of accuracy.
specified date	The specified date exactly. (No level of accuracy may be set.

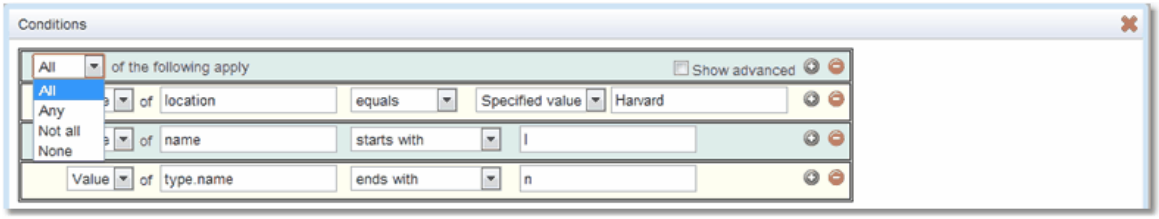
The **now** option means that the value of the *property* to be compared against is to be the date and time that the comparison is done. The **now plus** and **now minus** options indicate that the date to be calculated is relative to the time the comparison is done. When selecting either of these options, a further item is displayed to input how much time to add or subtract from the time of the comparison to reach the date-time that the *property* is being compared against. If the **specified date** option is selected, then a date-time entry is displayed to input the exact point in time to compare the *property* against.

The **Accuracy** is only shown for date or date-time properties, and it specifies how accurately the comparison is to be.

Then, when the *Condition* is checked, and **Calpendo** compares two dates, it first rounds both dates down to the year, month, day, hour, minute or second. After rounding both dates, it then compares the dates using the specified *relation*.

Combination Option

Once all the *conditions* have been created, choose how logically the *conditions* will work together (the combination option). Click on the drop down button in the top left corner of the *condition* box.



Combination Option	Description (For the combination to be TRUE)
All	All the individual *conditions* must be true. (Logical AND)
Any	At least one of the individual *conditions* must be true. (Logical OR)
Not All	As long as not all the individual *conditions* are true, this includes none of the *conditions* being true. (Logical NAND)
None	As long as none of the individual *conditions* are true. (Logical NOR)

This creates the following search *condition*:
Resources where (**location.name** starts with **h**) and (**type.name** equals **Room**) and (**requireCancellationReason** equals **True**)
At this point use **OK** to save the *condition* and **Cancel** to start again.

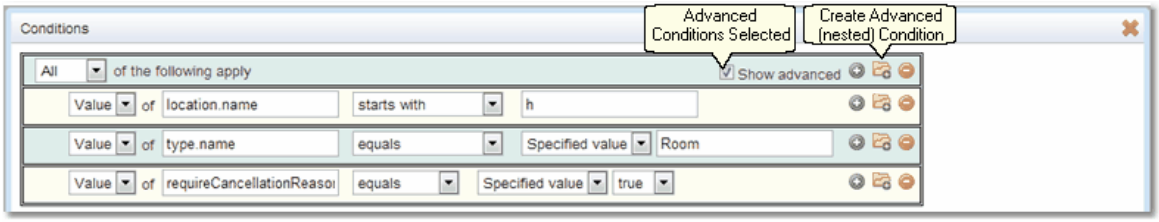
© 2010-2015 Exprodo Software

Nested Conditions

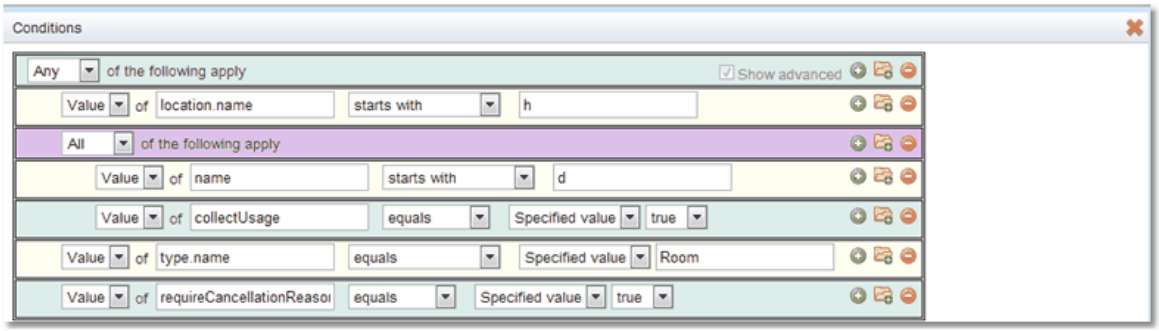
Nested *conditions* allow as many levels of *conditions* as the user requires. Each set of nested *conditions* also has its own combination option allowing the user to create as complex an overall *condition* as possible.

Each nested set of *conditions* is evaluated separately, and the overall result of the nested *condition* is then used as an individual *condition* result as part of the level above's *condition* calculation.

In order to use advanced *conditions* tick the **Show Advanced** box. Once ticked the **Create Advanced Condition** icon appears at the top of the *condition* list and on each of the current *conditions*.


Conditions
Advanced Conditions Selected
Create Advanced (nested) Condition
All of the following apply
Show advanced
Value of location.name starts with h
Value of type.name equals Specified value Room
Value of requireCancellationReaso equals Specified value true

Below the *conditions* have been expanded using a nested *condition*. The nested *condition* has two parts, **name** and **collectUsage** and it will be **True** if both sub *conditions* are **True**.


Conditions
Any of the following apply
Show advanced
Value of location.name starts with h
All of the following apply
Value of name starts with d
Value of collectUsage equals Specified value true
Value of type.name equals Specified value Room
Value of requireCancellationReaso equals Specified value true

This creates the following search *condition*:

Resources where (**location.name** starts with **h**) or ((**name** starts with **d**) and (**collectUsage** equals **true**)) or (**type.name** equals **Room**) or (**requireCancellationReason** equals **true**)

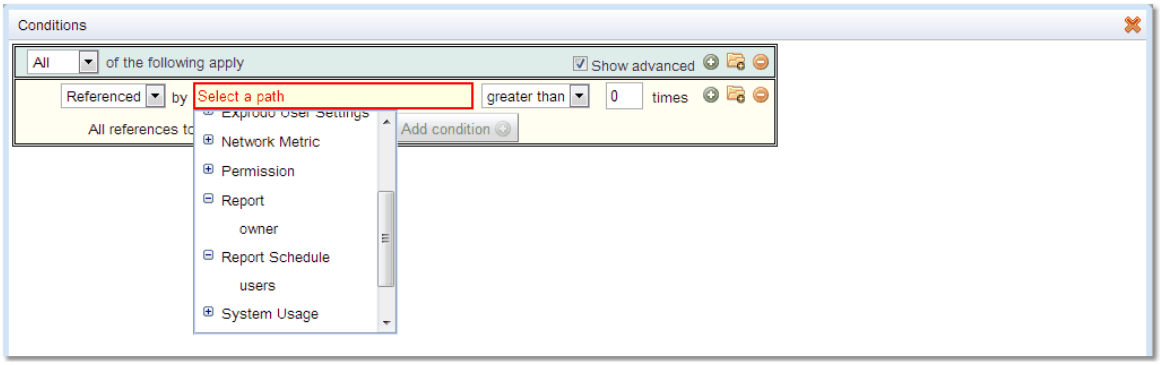
In order to test the *condition* the following steps will be used

1. Check the nested *condition*, if both *conditions* are **True** then the nested *condition* is **True** otherwise **False** (**All** *conditions* apply)
2. Check the top level *condition*, if any of the three top level *conditions* are **True** or the nested *condition* is **True** then the whole condition is **True** (**Any** *condition* applies)

© 2010-2015 Exprodo Software

Referenced

Ticking **Show advanced** allows the use of **Referenced** as a **Condition Type**. Using **Referenced** it is possible to search for *BiskitDefs* 446 that are referenced by other *BiskitDefs*. Using the **Relation Statement** and the **Value** this can be further refined to find those that are referenced a number of times. e.g. more than once, less than three times etc.



When using **Referenced** the application looks for all those *BiskitDefs* that have a *property* which points to the *BiskitDef* that is the object of the search. In this case the search *BiskitDef* is **Exprodo User** the application will give a drop down that lists all the *BiskitDefs* that have a *property* that points to an **Exprodo User** *BiskitDef*. Expanding a *BiskitDef* will list the *properties* that are available. For instance **Report** has an **owner** *property* and **Report Schedule** has a **users** *property*.

As an example to find all those users that have created **Reports**, set **Exprodo User** as the *BiskitDef* to be searched for and **Report.owner** (**owner** is a pointer to an **Exprodo User**) as the *property* defined in the **Property Path**, returned will be a list of **Exprodo Users** that currently have **Reports** in the **Report Manager**. If the **Relation Statement** and **Value** are set to **greater than 1 times**, then only those users with two or more reports will be returned.

See Also:

- Example Automatic Emails 306
- The examples in Permissions 278

© 2010-2015 Exprodo Software

3.10 Searching For Information

There are various ways to search for information in **Calpendo**, and several ways to display the information found. This section describes the generic search options that apply to any type of data.

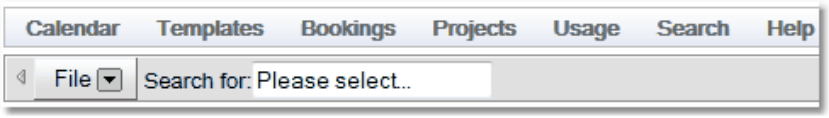
See Also

- Searching For Bookings 61 for how to search for information about *bookings* 446.
- Searching For Projects 72 for how to search for information about *projects* 448.
- Search For Usage 84 for how to search for information about *resource usage* 449.

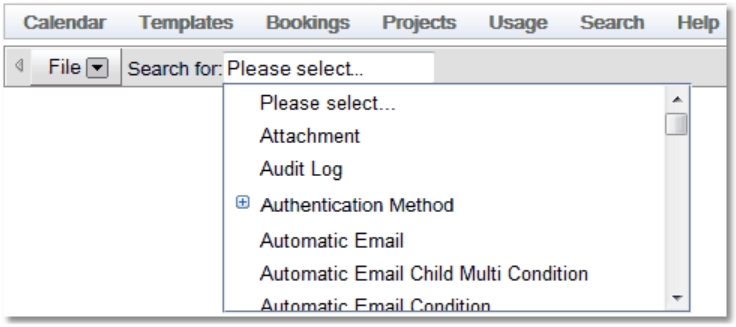
3.10.1 Search

The **Search** page allows searches for any type of data within **Calpendo**. By default, it appears under the **Search** menu, with a sub menu item also labelled **Search**. However, the menu may have been configured so that the **Search** page is not visible or in a different place.

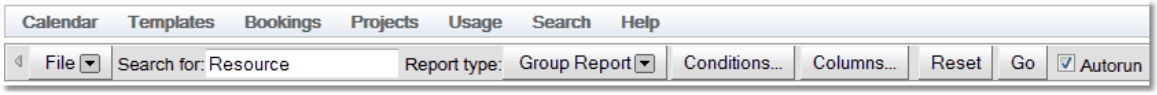
When you enter the **Search** page the following will be seen:



Choose which *Biskit Type* 446 to search for. Clicking on **Please Select** will provide a drop down menu with the complete list of types available to search.



Once a *Biskit Type* is selected additional options will appear.



There are four report types: *List* 447, *Summary* 449, *Single Item* 449 and *Group* 447. The user can also add *conditions* 447 to the search (not *Single Item*), change the columns that are going to be viewed in the returned search (not *Summary* or *Single Item*), or reset the search *conditions* to the default (not *Single Item*).

This icon will toggle whether the **File** and **Report Type** options are available.

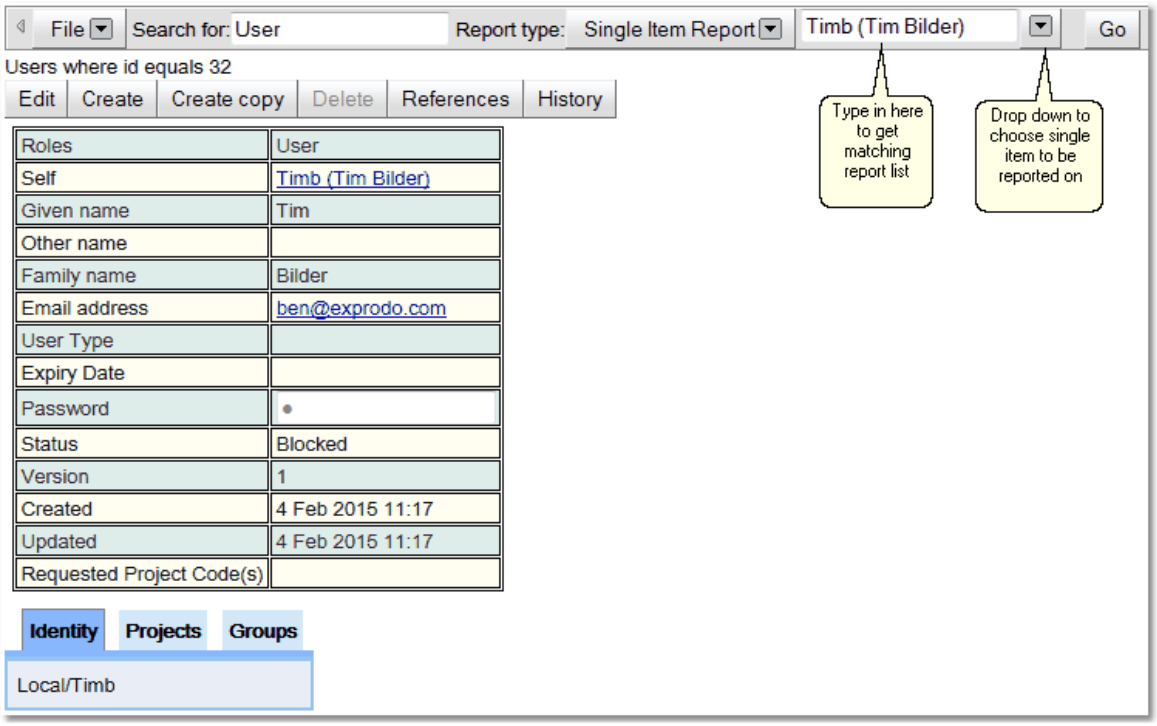
The **Go** button will run the current search set up.

© 2010-2015 Exprodo Software

Autorun if selected will automatically run new search set ups as they change, so changing **Columns**, ***Conditions***, **Report Type** and even what is being searched for, will get the search to be re-run as each setting is changed. If searches are being done on large amounts of data or the user needs to make multiple changes between search runs it is more efficient if **Autorun** is not selected. If at any time having made changes and **Autorun** is selected but a new search has not run then press the **Go** button.

3.10.1.1 Single Item Report

Shows the result of the search as a single record.



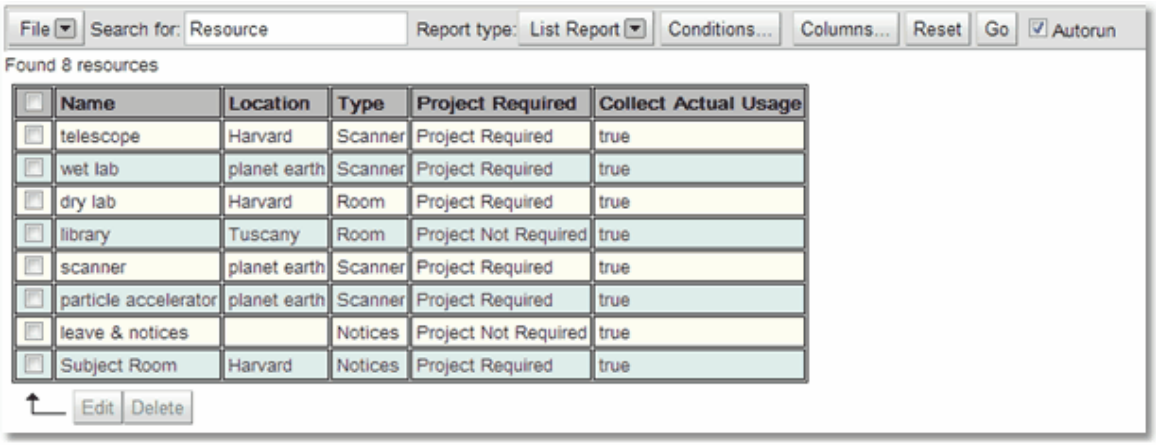
Either use the drop down to get a full list of possible options to be scrolled through or type in the input box to get a shortened list of possible matching objects.

If the user has the correct *permissions* [448] they may edit the individual records or make changes to multiple records. All of this will be covered in the chapter on Editing Search Information [117]

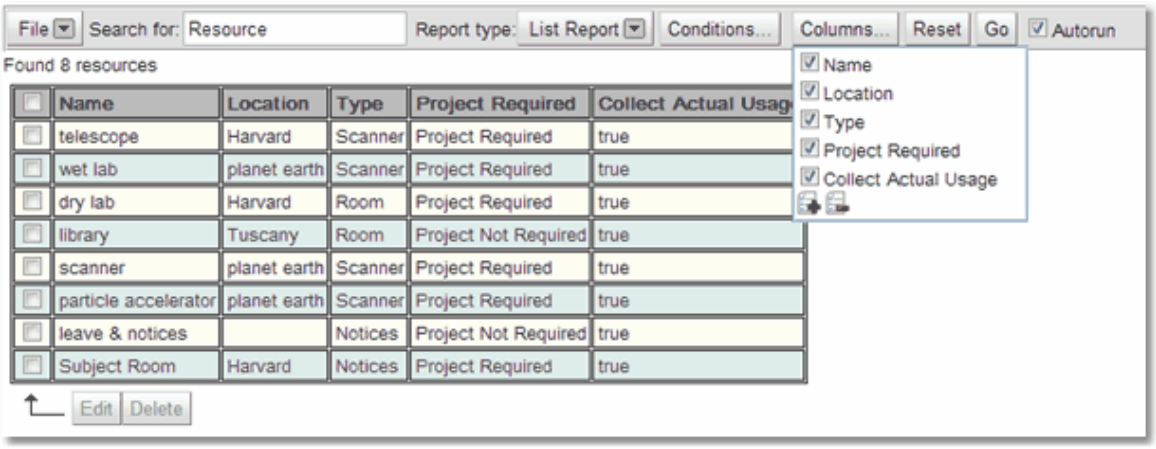
© 2010-2015 Exprodo Software

3.10.1.2 List Report

Shows the result of the search as a list of records

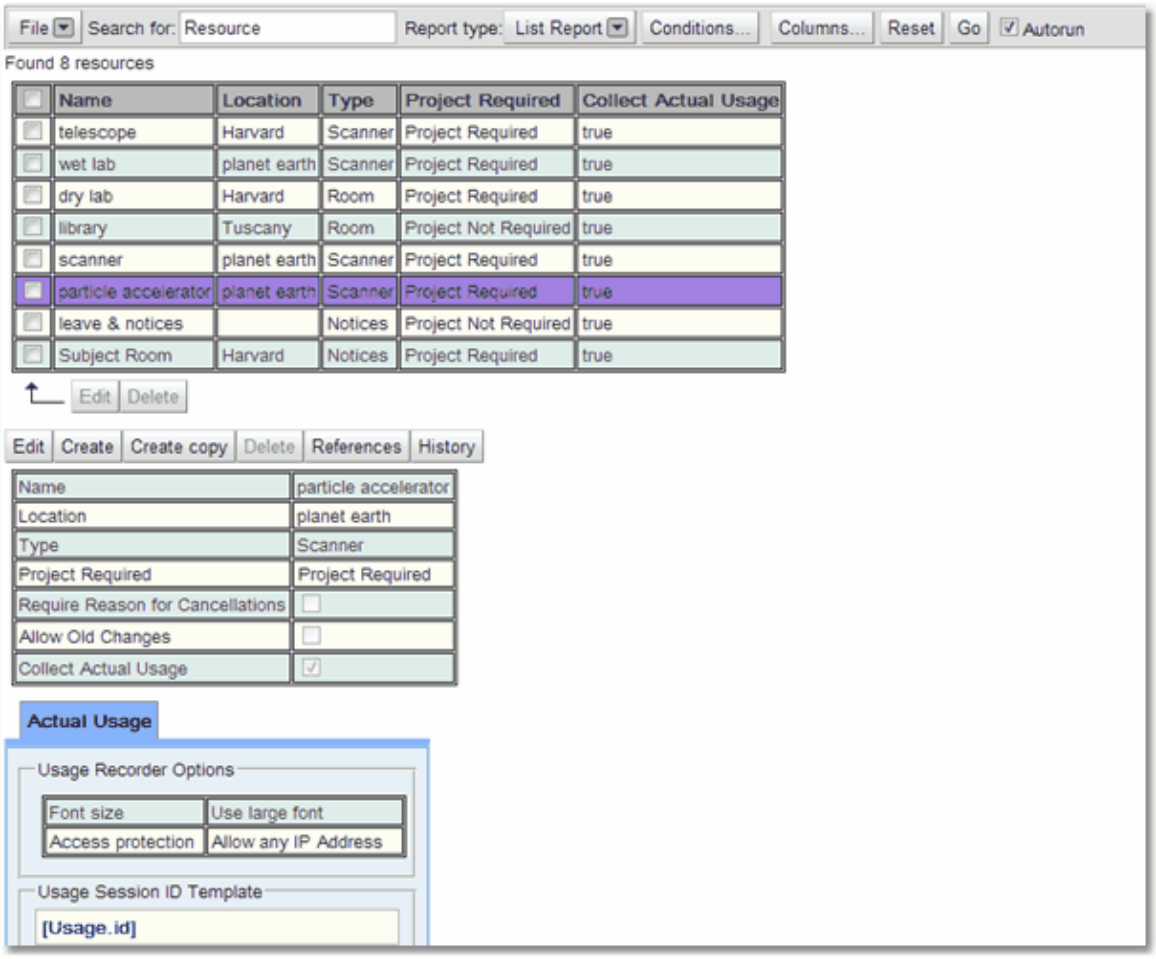


Specify which columns are viewed by clicking on the **Columns...** button and get a drop down list of tick box's for all available *properties*[448], those who have their boxes ticked will be shown.



© 2010-2015 Exprodo Software

If an individual record is clicked an expanded view of that record will appear below the list.



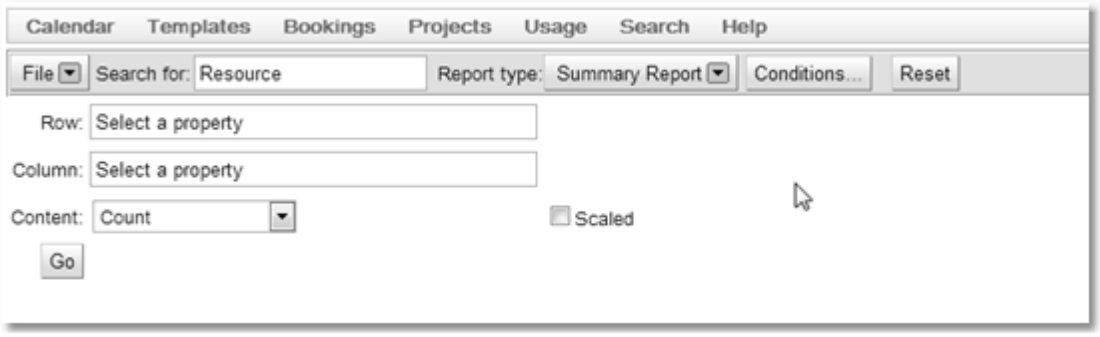
If the user has the correct *permissions* [448] you may edit the individual records or make changes to multiple records. All of this will be covered in the chapter on Editing Search Information [117].

© 2010-2015 Exprodo Software

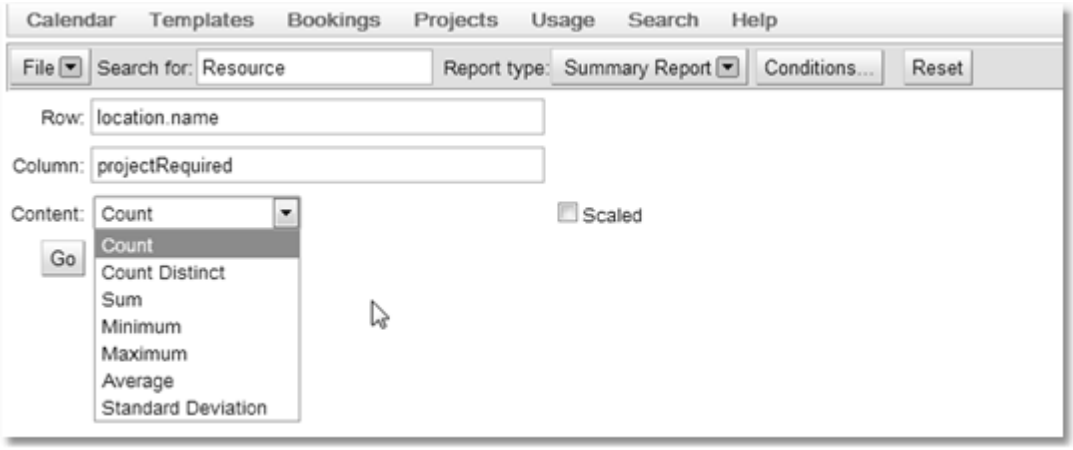
3.10.1.3 Summary Report

Shows the report as a table, where the user can define the *properties*[448] that are used for the rows and the columns of the table, and how the content of each cell will be calculated.

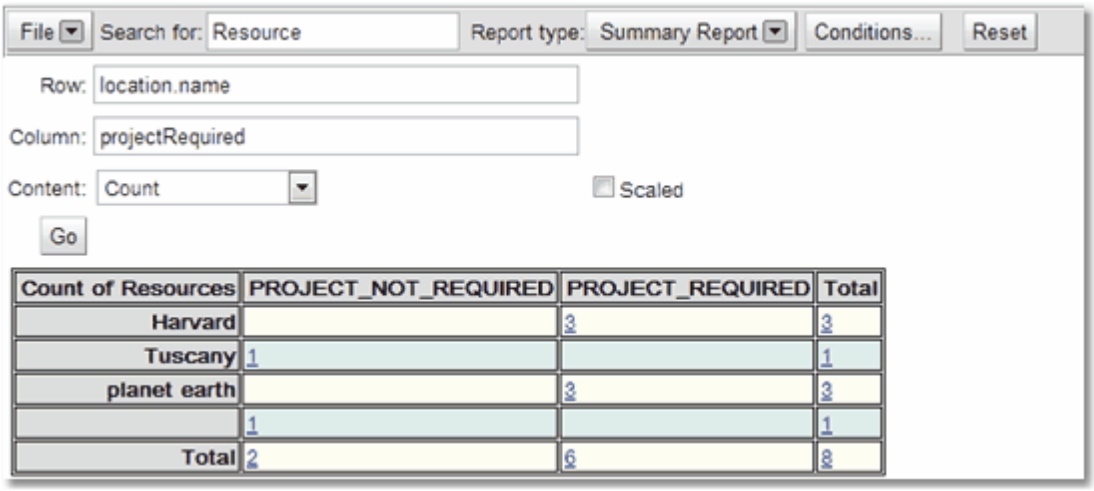
When the *Summary Report*[449] view appears the user will be asked to select the *properties* that will be used for the rows and for the columns.



Once the *properties* have been selected, then select how the content is to be displayed.



The default content display is count, this will display the content as a simple count of how many records match the row/column combination, with totals for each row and column.

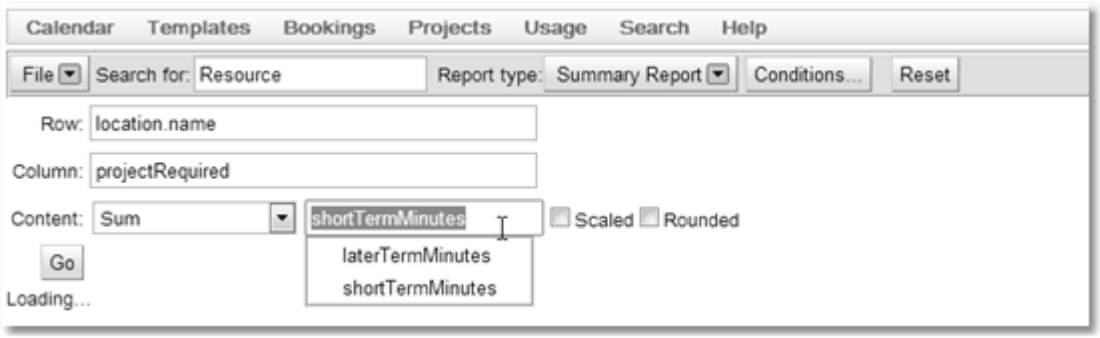


Count of Resources	PROJECT_NOT_REQUIRED	PROJECT_REQUIRED	Total
Harvard		3	3
Tuscany	1		1
planet earth		3	3
	1		1
Total	2	6	8

© 2010-2015 Exprodo Software

Complex Content Types

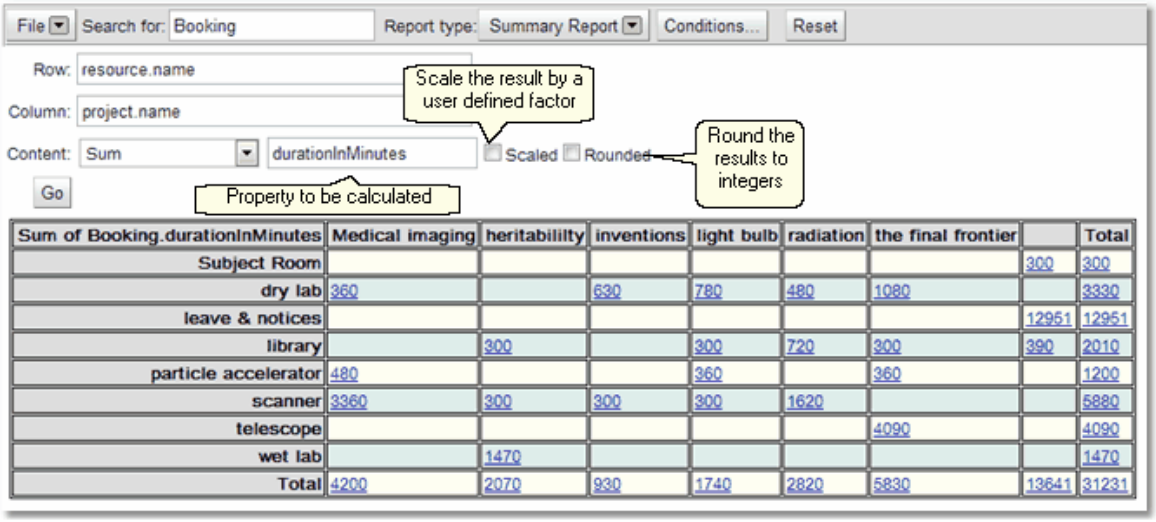
All the other content types require a numerical or date property as an additional parameter (the exception is **Count Distinct** which will work with any type) . The value of this parameter will be used to determine the value stored in each cell.



Content Type	Description
Count Distinct	How many different values to the *property* are there for this cell.
Sum	The total of this *property* for this cell (numerical only)
Minimum	The smallest value for this *property* for this cell
Maximum	The largest value of this *property* for this cell
Average	The average value of this *property* for this cell
Standard Deviation	The standard deviation of the values of this *property* for this cell.
Value At Minimum Of	Choose a property to find the minimum of, and the value shown in this column will be the value of the current property when the property being compared with is at a minimum.
Value At Maximum Of	Choose a property to find the maximum of, and the value shown in this column will be the value of the current property when the property being compared with is at a maximum.

Below is an example of a table showing the total booking duration in minutes for each *project* 448 for each resource

© 2010-2015 Exprodo Software

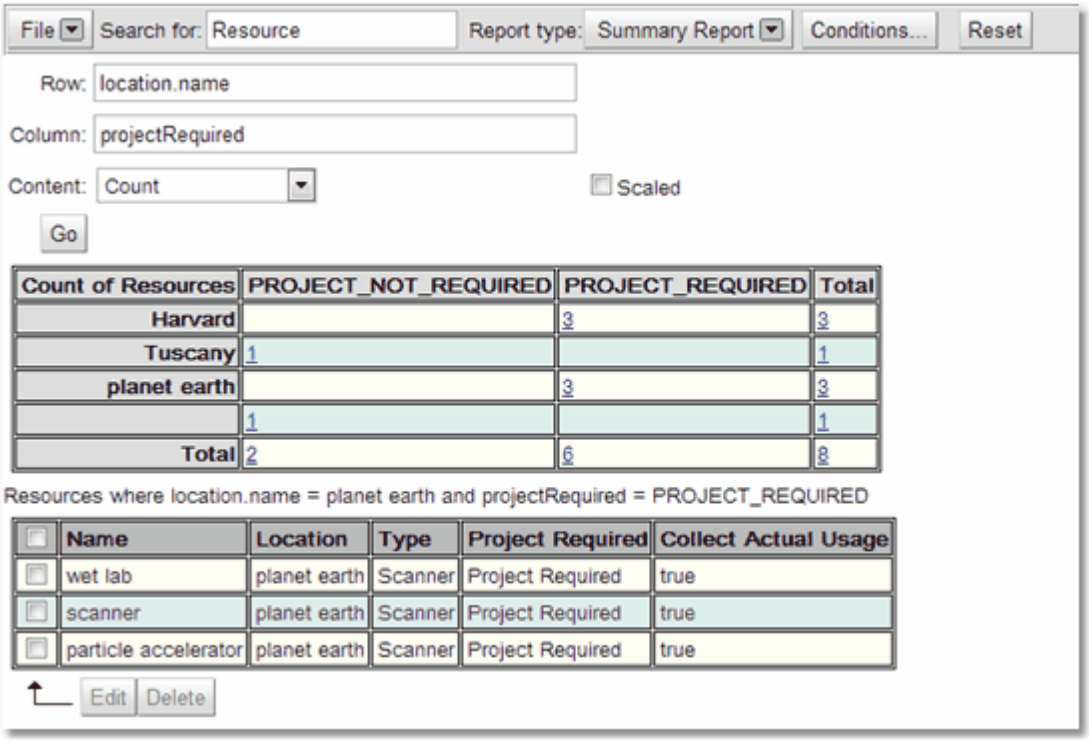


Sum of Booking.durationInMinutes	Medical imaging	heritabililty	inventions	light bulb	radiation	the final frontier		Total
Subject Room							300	300
dry lab	360		630	780	480	1080		3330
leave & notices							12951	12951
library		300		300	720	300	390	2010
particle accelerator	480			360		360		1200
scanner	3360	300	300	300	1620			5880
telescope						4090		4090
wet lab		1470						1470
Total	4200	2070	930	1740	2820	5830	13641	31231

Use the **Rounded** option to round numbers to integers
Use the **Scaled** option to scale the results by the user requested factor (i.e if the scaling factor is 10 divides all results by 10).

Viewing Records

Records associated with each cell can be viewed by clicking on the number in the cell. This will give a *List Report*[447] of the records for that cell beneath the table.



Count of Resources	PROJECT_NOT_REQUIRED	PROJECT_REQUIRED	Total
Harvard		3	3
Tuscany	1		1
planet earth		3	3
	1		1
Total	2	6	8

Resources where location.name = planet earth and projectRequired = PROJECT_REQUIRED

Name	Location	Type	Project Required	Collect Actual Usage
wet lab	planet earth	Scanner	Project Required	true
scanner	planet earth	Scanner	Project Required	true
particle accelerator	planet earth	Scanner	Project Required	true

As with the *List Report*[110] clicking on an individual record will give an expanded view of that record below the list.

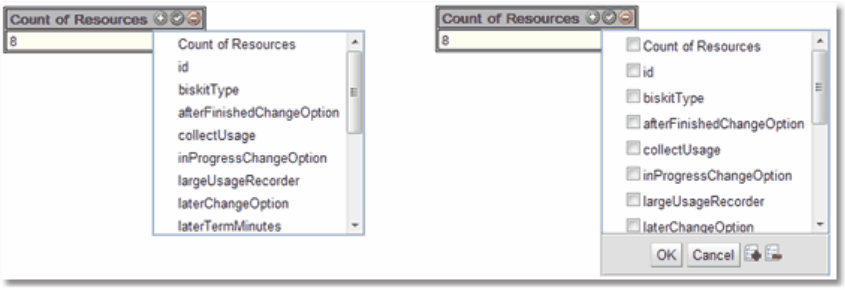
If the user has the correct *permissions*[448] they may edit the individual records or make changes to multiple records. All of this will be covered in the chapter on Editing Search Information[117]

© 2010-2015 Exprodo Software

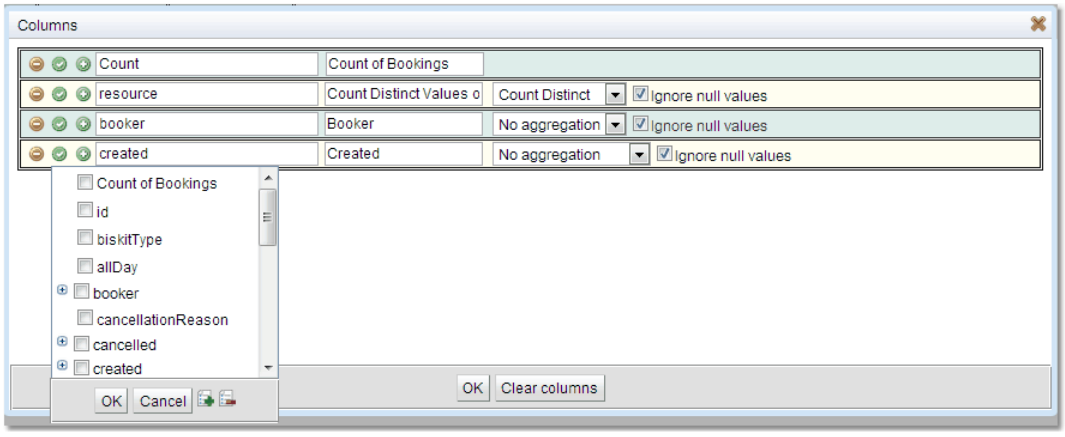
3.10.1.4 Group Report

Shows the report as a table with one column showing a **Count of Biskit Type**. The table can then be extended to show other columns for *properties*[448] available within the *Biskit Type*[446] chosen.

To choose more columns either use the **Columns...** button on the toolbar or the **green +** (to add one column), **red -** (to delete the current column), **green tick** (to add multiple columns, tick all columns that are required and then click **OK**, or use the **green +** icon on the menu bar to select all the columns or the **red -** icon to deselect all the options).

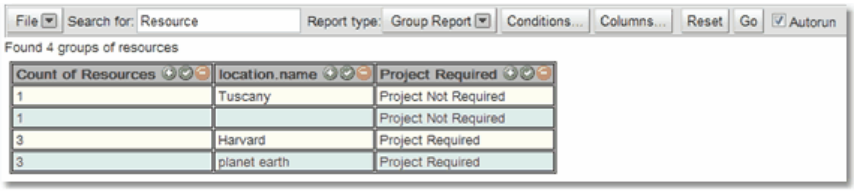


If the **Columns...** button is clicked a pop up appears, in which the columns can be chosen using the same three buttons as described above. At the same time the user can decide what aggregation to use, **No aggregation** or **Count Distinct**, as well as whether to **Ignore null values**.



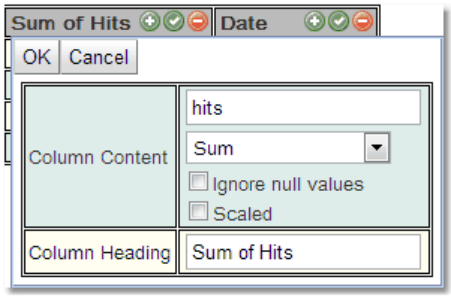
© 2010-2015 Exprodo Software

Once the columns have been chosen the table will redisplay using the appropriate number of rows to display the records, grouping records by the columns chosen giving a count in the first column.



Once the columns are displayed, move the cursor over the name of a column and a pop up will appear with **Change** in it. Click on **Change** in order to change

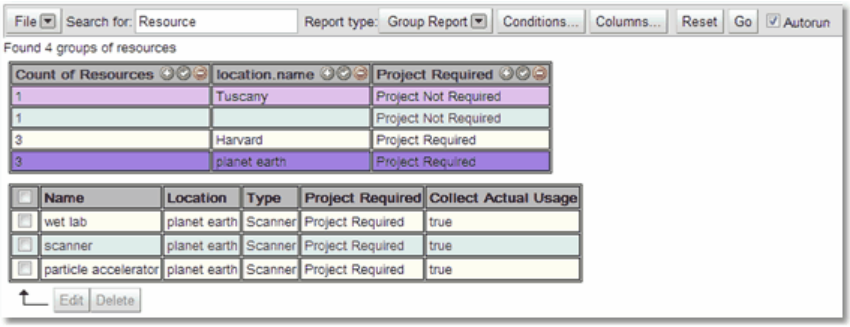
1. The column header
2. The column content



The column content can be changed to

1. Another *property*.
2. A complex content type, see Complex Content Type[113] in the Summary Report[112] section for more details on the types available.
3. Whether null values are ignored.
4. Whether the result should be scaled see Complex Content Type[113] in the Summary Report[112] section for more details on scaling.

Clicking on a cell will produce a *List Report*[447] of the records associated with that cell below the table.



As with the *List Report*[110] clicking on an individual record will give an expanded view of that record below the list.

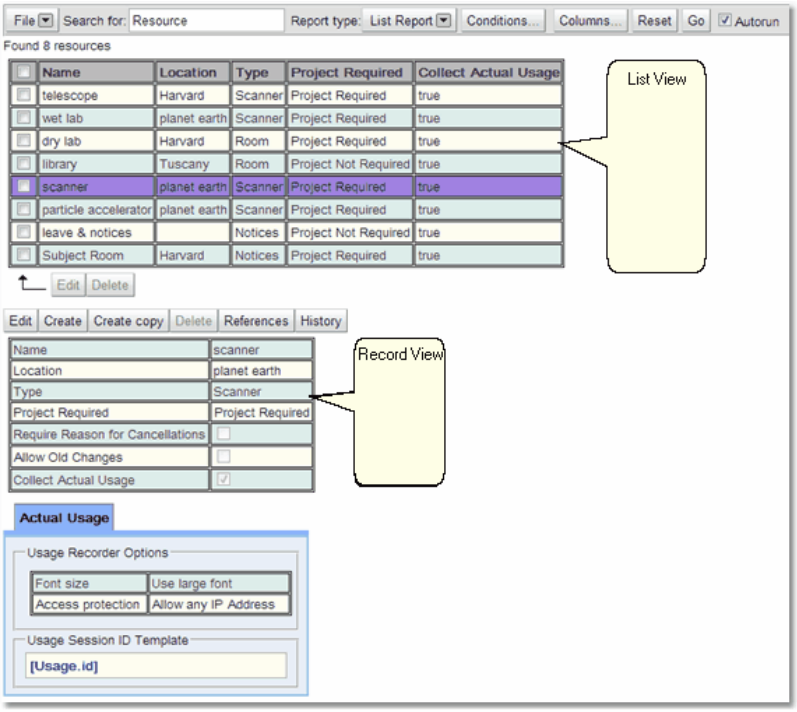
If the user has the correct *permissions*[448] they may edit the individual records or make changes to multiple records. All of this will be covered in the chapter on Editing Search Information[117]

© 2010-2015 Exprodo Software

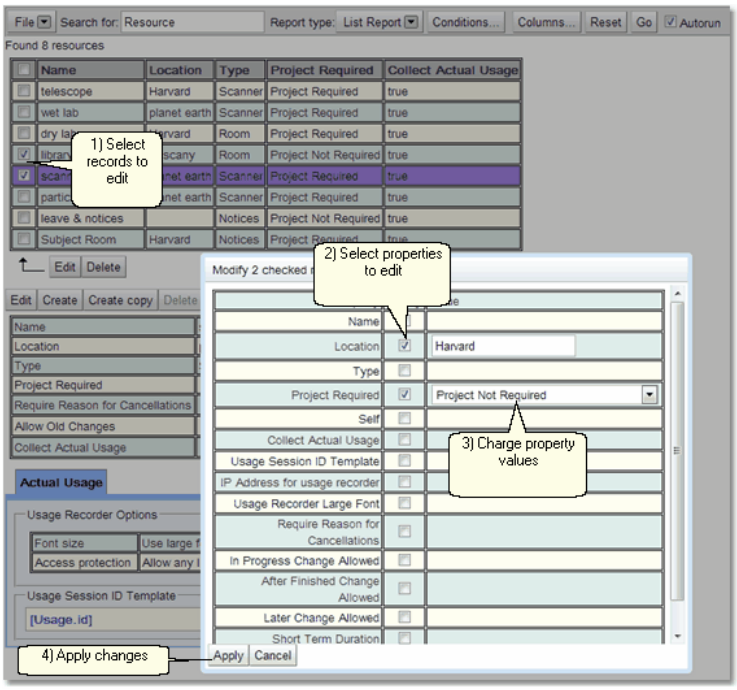
3.10.1.5 Editing Search Information

Once the search information is in either a *List Report* [447] view or as an individual record the information can be edited if the user has the *permissions* [448] to do so. The administrator will have set up the *permissions* for your **Calpendo**.

For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition [96] chapter.

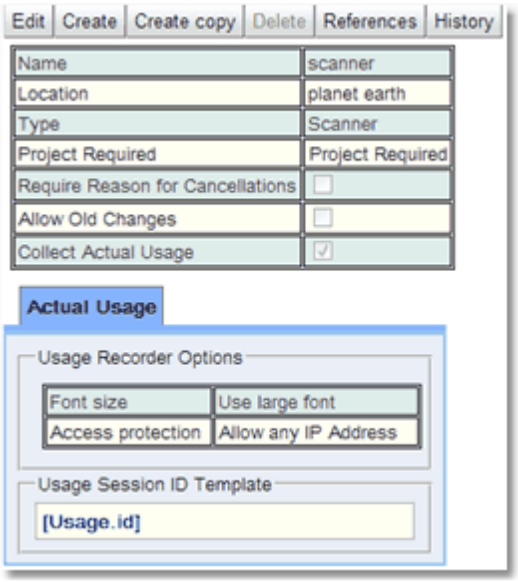


For a more complete explanation on how to edit records in the list view read the How to Edit Multiple Items At Once [128] section of the Data Explorer [127] chapter.

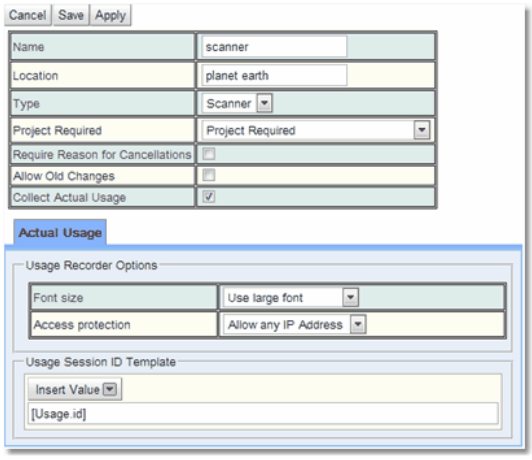


© 2010-2015 Exprodo Software

For a more complete explanation on how to edit records in the record view read the How to Edit A Single Item[130] section of the Data Explorer[127] chapter.



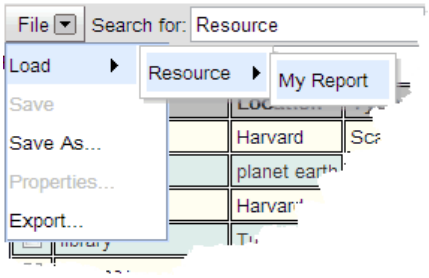
Once in edit mode the changes can be made.



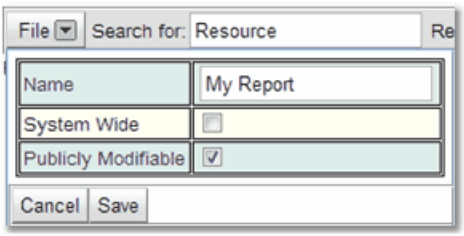
© 2010-2015 Exprodo Software

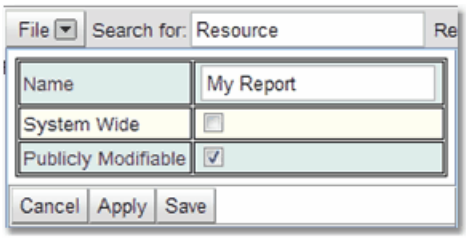
3.10.1.6 Saving And Reusing Searches

Once a search has been created that is useful the user may wish to save it for later use. This can be done using the drop down button from the **File** option on the toolbar and choosing **Save As**. This will save the search as a report accessible from the **Search->Reports**[121] page. Using the **File** option the user can also load saved searches to run them, edit the *properties*[448] of any saved searches and export search information to a file called **report.csv**.



When a search has been created use the **File->Save As** to save the search. Give it a **Name**, decide whether the search is accessible to all users (**System Wide**) and whether anyone can edit the search (**Publicly Modifiable**). Then save the search.





When a search is saved it can be reloaded using **File->Load**. If a search is loaded then the *properties* can be edited using the **File->Properties** menu option. When edited they can be saved. In this case **Save** and **Apply** do the same thing, that is save the changed information and leave edit mode.

© 2010-2015 Exprodo Software

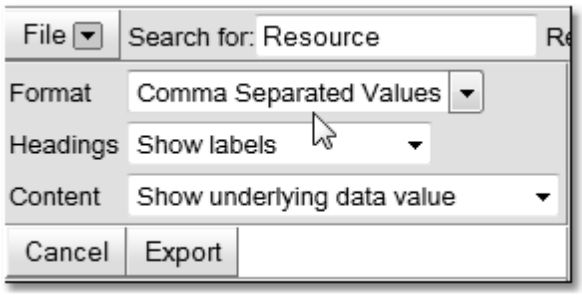
Exporting Search Information

A user can export any data they have found through a search to a .csv, a tab separated file or native **Excel** file (either pre 2007 xls or post 2007 xlsx) which can then be loaded into such applications as **Excel**.

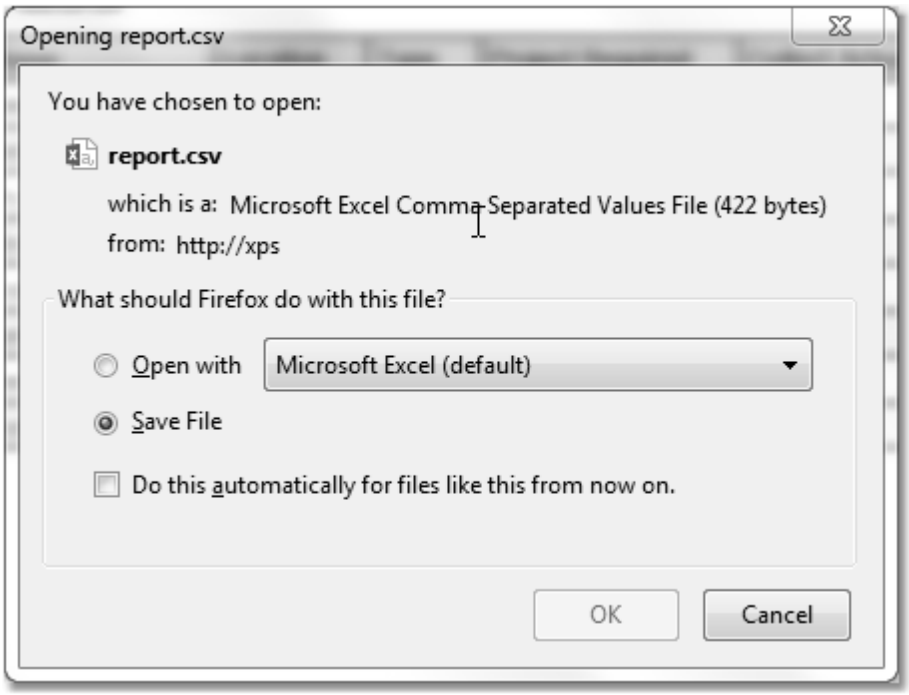
Using **File->Export** the current search data will be exported to a file.

There are three options:

1. How the file will be formatted, choose between **Comma Separated Values, Excel xls**, **Excel xlsx** and **Tab Separated Values.**
2. What headings will be used, these can be **Show labels** or **Show property names**. Labels are those names that have been assigned by the administrator to a *Biskit Type* 446, the *property* name is the name of the *property* in the database.
3. Whether enumerated types have their names shown, **Show value labels** where defined or the value as stored in the database, Show underlying data value.



When the format for the export is chosen press **Export**, then choose whether to open the exported information directly into another application, or whether to save the file. If the file is saved it will be called **report.csv** and found in the users **Downloads** folder.



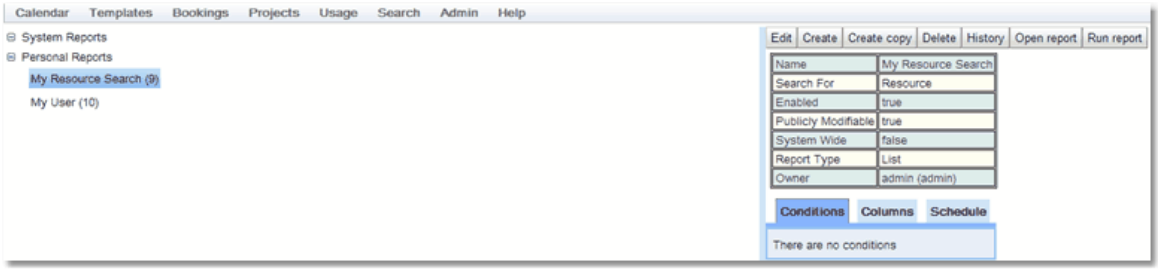
© 2010-2015 Exprodo Software

3.10.2 Report Manager

The **Report Manager** page allows the management of reports within **Calpendo**. By default, it appears under the **Search** menu, with a sub menu item labelled **Reports**. However, the menu may have been configured so that the **Reports** page is not visible by the administrator.

Once useful searches have been created they can be saved for future use as reports (see Saving and Reusing Searches 119). The reports page allows you to view, manage and schedule your reports as well as creating new ones.

The page is split into two. On the left is a list of the **System Reports** (those that anyone can see) and the users **Personal Reports**. On the right is the area where to edit, schedule and create reports, when the user enters the **Reports** page this area is initially empty. Select a report in the left hand pane to see the properties 448 of that report in the right hand pane. Where the page splits can be moved to change the space available for each of the sides.



A report has a number of basic properties:

Property	Description
Name	The name of the report as assigned by the user.
Search For	The *biskit type* 446 that will be searched for.
Enabled	If the report will run if scheduled.
Publicly Modifiable	If a system wide report then anyone can modify it.
System Wide	Can everyone see the report or is it a personal one. If this is true the report will appear under **System Reports** rather than **Personal Reports**.
Report Type	Will the report display as *List* 447, *Summary* 449 or *Group* 447 report
Owner	The user who created the report and if **Publicly Modifiable** is false, the only person who can edit a report if it is also **System Wide**

and tabs to see other properties:

Tab	Description
Conditions	The *conditions* 447 (if any) associated with the report. (see the Setting Search Conditions chapter for more information about *conditions* and how to edit them)
Columns	The columns that will be displayed in the report
Schedule	Whether the report is scheduled to run and, when it will run, who the report will be e-mailed to and the format of that e-mail.

© 2010-2015 Exprodo Software

The Menu Bars

The menu bar has two variations. In **View** mode:

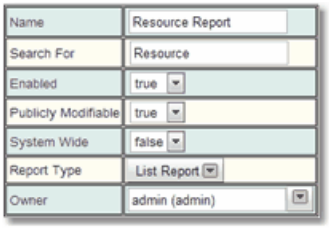
after you press the **Edit** button and enter edit mode:

For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition 96 chapter.

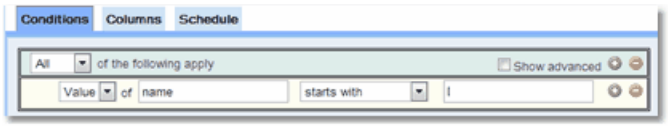
Report Menu Bar	Description
Open Report	Opens the **Search** page in a state ready to run your report. Click **Go** to run the report. This will be greyed out in edit mode
Run report	Opens the **Search** page and runs the report. This will be greyed out in edit mode

Editing Reports

Once a report is in edit mode the basic *properties* can be changed in the top section, all have drop down menus for the options available other than the report name which can be overwritten.

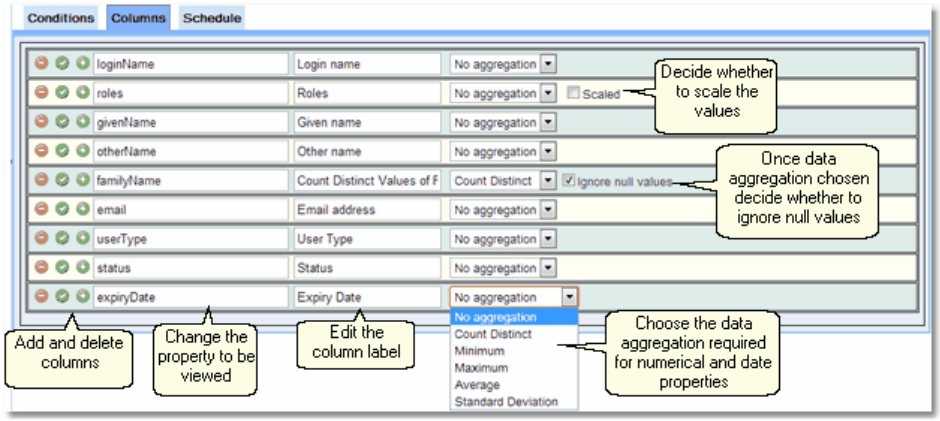


Use the tabs to add or change the *conditions* (see the Setting Search Conditions chapter for more information about *conditions* and how to edit them).



© 2010-2015 Exprodo Software

Use the tabs to change the columns that are viewed, what is viewed in them and how they are labelled.



The first column allows the user to add or delete columns from the report, the **green +** (to add one column), **green tick** (to add multiple columns, tick all columns required to be added then click **OK**), **red -** (to delete the current column). (see the Group Report[115] chapter for more information on how they work)

The second column allows the user to change the *properties* that will be viewed in a column.

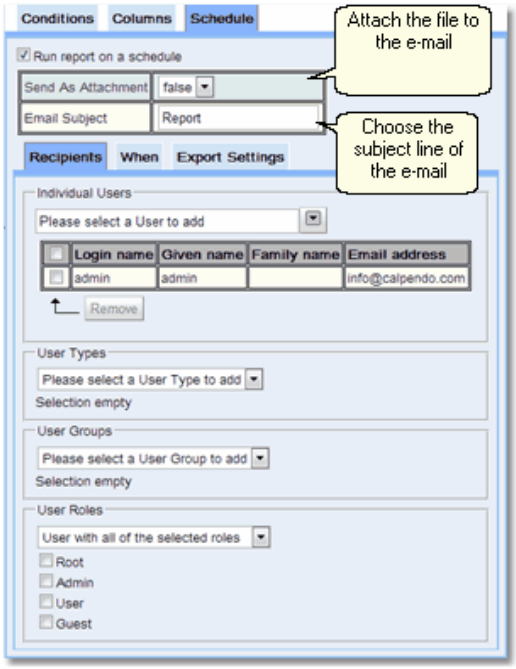
The third column allows the user to change the label at the top of the column in the report.

The fourth column allows the user to decide whether absolute values or data aggregation is required. See Summary Report[112] chapter, Complex Content Types[113] section for more information. If aggregation of the data has been chosen, then decide whether to ignore null (empty or 0) values.

© 2010-2015 Exprodo Software

Scheduling Reports

If box marked **Run report on a schedule** is ticked, options allowing for the setting up of the scheduling of the report appear.



Choose whether to attach the report as a file to the email or whether the body of the email is the report data.

Choose the subject line to appear on the email.

Select the **Recipients** tab to decide who to send the email to.

Select the individual users the mail is to go to.

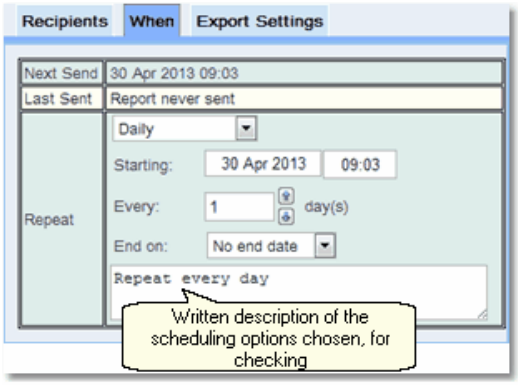
Select the *User Types* 450 the mail is to go to. The mail will be sent to all users of these types.

Select the *User Groups* 450 the mail is to go to. The mail will be sent to all users in these groups.

Select the *User Roles* 450 the mail is to go to. The mail will be sent to all users with these roles.

Emails will only be sent to an address once. So if multiple users have the same email address or with your selections of **Individual,** *Type*, *Group* or *Role* you would have accessed a user multiple times, they will only get one email.

Select the **When** tab to schedule the report.



Next Send: The time and date the mail will next be sent with the current settings.

Last Sent: The time and date the report was last sent.

Select the frequency of the mail: Daily, Weekly, Monthly by Date, Monthly by Day, Yearly

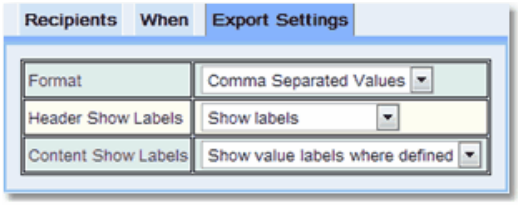
Select when the emails will start

Specify the frequency: every other day, every third week etc.

Select when the emails will end if required.

© 2010-2015 Exprodo Software

Select the **Export Settings** tab to set up the format of the exported file.



1. **Format**: How the file will be formatted, choose between **Comma Separated Values, Excel xls**, **Excel xlsx** and **Tab Separated Values.**

1. **Header**: What headings will be used, these can be **Show labels** or **Show property names**. Labels are those names that have been assigned by the administrator to a *biskit type*, the property name is the name of the *property* in the database.
2. **Content**: Whether enumerated types have their names shown, **Show value labels where defined** or the value as stored in the database, **Show underlying data value**.

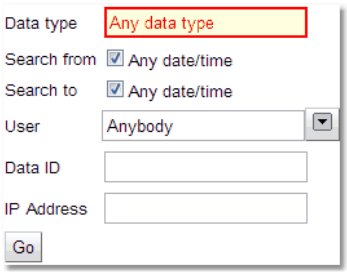
© 2010-2015 Exprodo Software

3.10.3 History

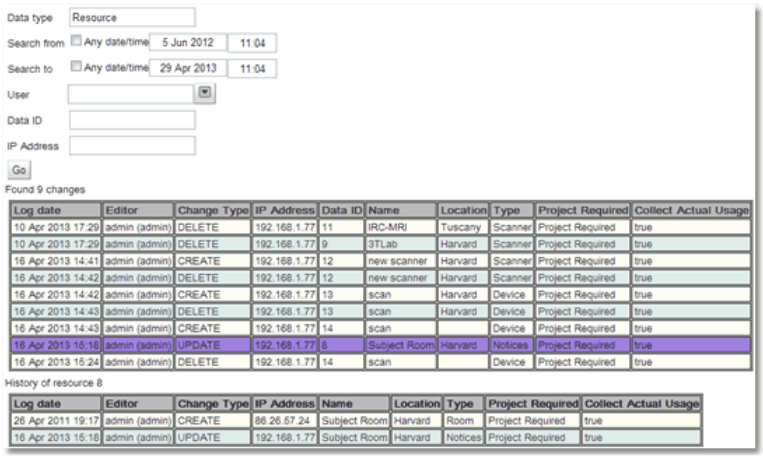
The **History** page allows the user to look into the audit log kept by **Calpendo** that shows what has been changed, when and by whom. This page is normally found on the **Search** menu as **Search-->History**. However, the menu may have been configured so that the **History** page is not visible.

To use the page, first enter the search criteria, and then press the **Go** button to see what matching changes can be found. The search criteria can be:

- **Data type**, choose what type of *Biskit*[446] has been changed. For example, if the user is interested in changes to *bookings*[446], then select **Booking**. If a *Biskit Type*[446] is not selected, then the search will include changes made to any type of *Biskit*.
- **Search from** and **Search to**, to restrict the search to particular periods of time. It is highly recommended to put limits on the time period using these options so that the amount of returned results are limited.
- **User**, choose to see changes made only by one particular user, or by anybody.
- **Data ID**, choose the unique ID number to search for. For example, if the user is interested in a *project*[448], and they know its ID is 42, then enter 42 here to restrict results to changes made to project 42.
- **IP Address**, search by changes made from a particular internet address.



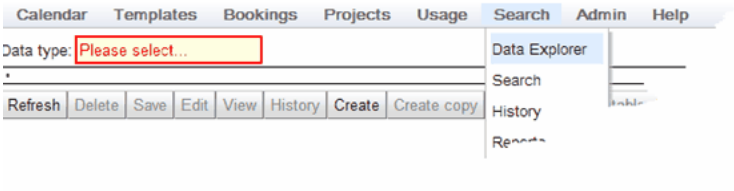
After pressing **Go**, a list of matching changes is displayed. Each one will represent a change to something within **Calpendo**. When the list appears, click on one of them to see all changes to the same *Biskit*. For example, if the search was for changes to *resources*[449] made on a particular day, returned will be a list of changes that affected all the *resources* on that day. By clicking on one of them, the user would then see all changes made to that *resource*, regardless of when that change was made.


Data type
Resource
Search from
Any date/time
5 Jun 2012
11:04
Search to
Any date/time
29 Apr 2013
11:04
User
Data ID
IP Address
Go
Found 9 changes
Log date
Editor
Change Type
IP Address
Data ID
Name
Location
Type
Project Required
Collect Actual Usage
10 Apr 2013 17:29 admin (admin) DELETE 192.168.1.77 11 IRC-MRI Tuscany Scanner Project Required true
10 Apr 2013 17:29 admin (admin) DELETE 192.168.1.77 9 3TLab Harvard Scanner Project Required true
16 Apr 2013 14:41 admin (admin) CREATE 192.168.1.77 12 new scanner Harvard Scanner Project Required true
16 Apr 2013 14:42 admin (admin) DELETE 192.168.1.77 12 new scanner Harvard Scanner Project Required true
16 Apr 2013 14:42 admin (admin) CREATE 192.168.1.77 13 scan Harvard Device Project Required true
16 Apr 2013 14:43 admin (admin) DELETE 192.168.1.77 13 scan Harvard Device Project Required true
16 Apr 2013 14:43 admin (admin) CREATE 192.168.1.77 14 scan Device Project Required true
16 Apr 2013 15:18 admin (admin) UPDATE 192.168.1.77 8 Subject Room Harvard Notices Project Required true
16 Apr 2013 15:24 admin (admin) DELETE 192.168.1.77 14 scan Device Project Required true
History of resource 8
Log date
Editor
Change Type
IP Address
Name
Location
Type
Project Required
Collect Actual Usage
26 Apr 2011 19:17 admin (admin) CREATE 86.26.57.24 Subject Room Harvard Room Project Required true
16 Apr 2013 15:18 admin (admin) UPDATE 192.168.1.77 Subject Room Harvard Notices Project Required true

© 2010-2015 Exprodo Software

3.10.4 Data Explorer

The **Data Explorer** page allows examination of the data within **Calpendo**. By default, it appears on the menu here:

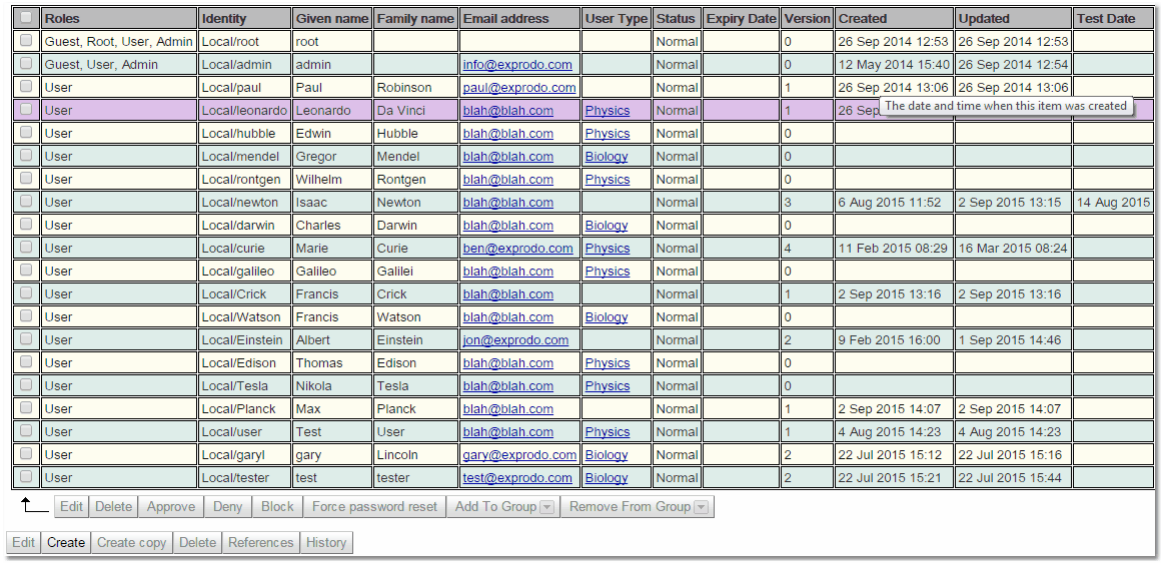


Select a *Biskit Type* 446, and a table appears that shows all *Biskits* 446 of that type.

> Note: The **Data Explorer** page does not yet handle well the situation when there is lots of *Biskits* of the selected type. It is highly recommended that the user does not attempt to use the **Data Explorer** to view *Bookings* 446, for example, since there may be many of them.

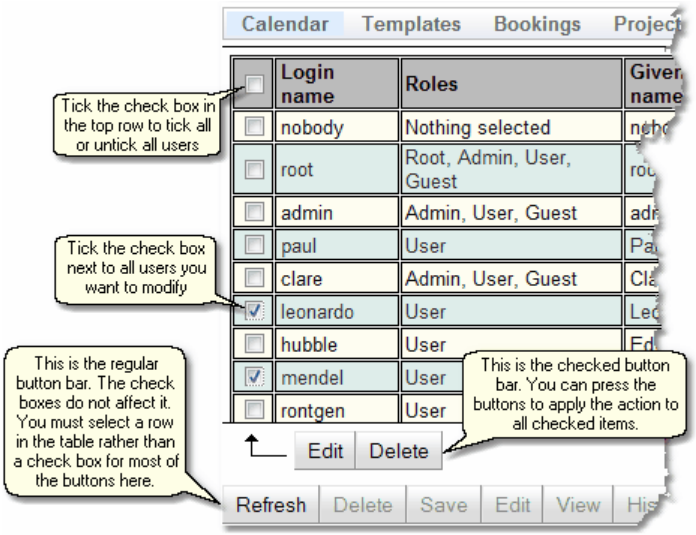
Apart from being able to choose the type of data to look at, the **Data Explorer** page is identical to several other pages in **Calpendo**, such as the list of users shown in Modifying Users.

Select **User** as the *Biskit Type*, then a list of users will appear and the user will be able to examine or modify them (*permissions* 278 permitting).

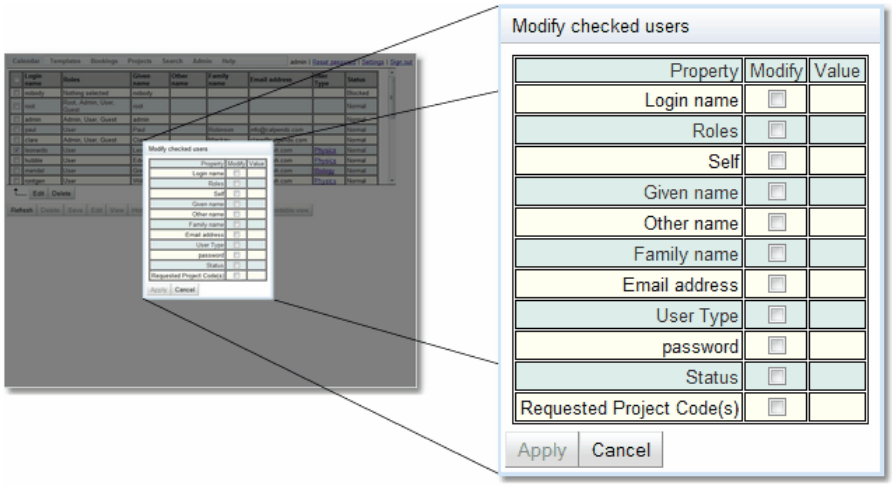


How To Edit Multiple Items At Once

You can edit many items in one go quite easily, using the check boxes and the checked button bar as follows:

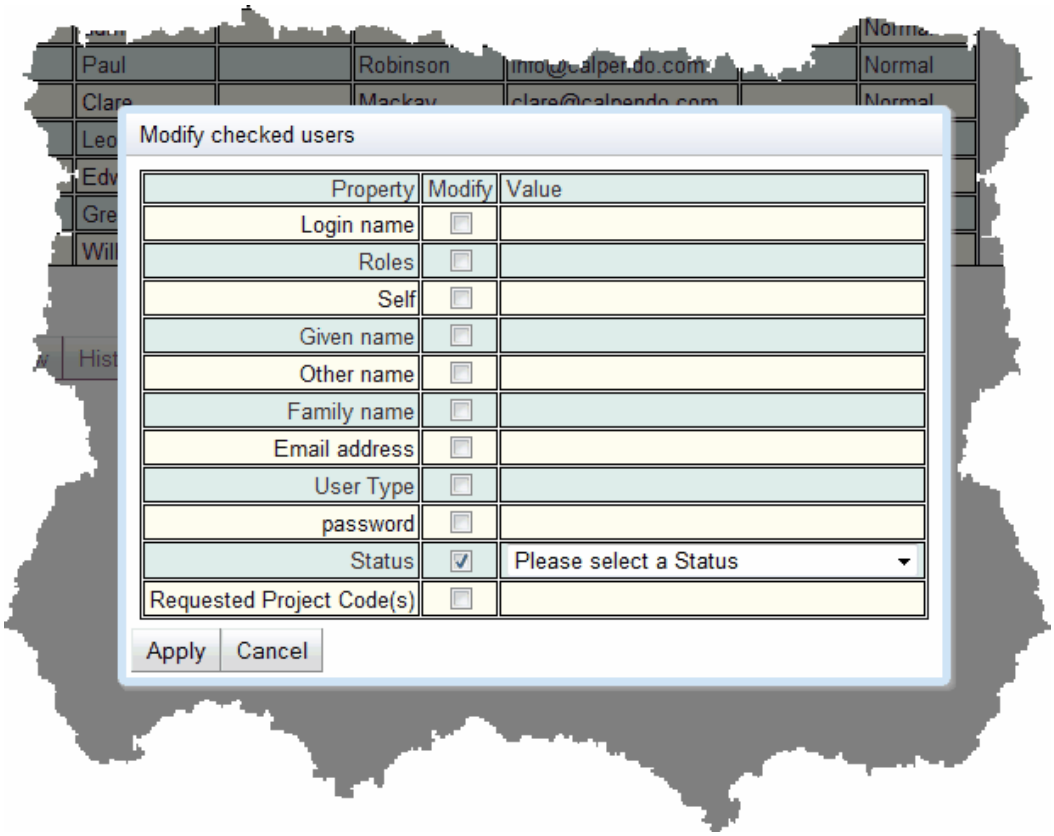
1. Tick the check boxes next to each item to be changed, or tick the box in the header row to tick all boxes.
2. 

3. Once one or more check boxes have been ticked, the checked button bar will change so it is no longer greyed out. Press the **Edit** button to edit all the ticked items.
4. When you press the **Edit** button, a pop-up appears showing most of the *properties*[448] that are defined for the selected *Biskit Type*. However, some may be missing if they are too complex to edit in this pop-up (for example, *properties* that contain lists of things). If these properties are **Grouped** or have a **Layout** defined then the pop up will be hierarchical.



© 2010-2015 Exprodo Software

5. Tick the check boxes of one or more *properties* shown in the pop-up. Once done, the **Apply** button changes so it is no longer greyed out, and an editor appears in the **Value** column:



6. Choose a value for each *property* whose check box you tick. When finished, press the **Apply** button. Every ticked item will then be modified, with only the ticked *properties* being changed. Whatever values selected for each *property* will be applied to all the ticked items. *Properties* that are not ticked are not affected by the change.

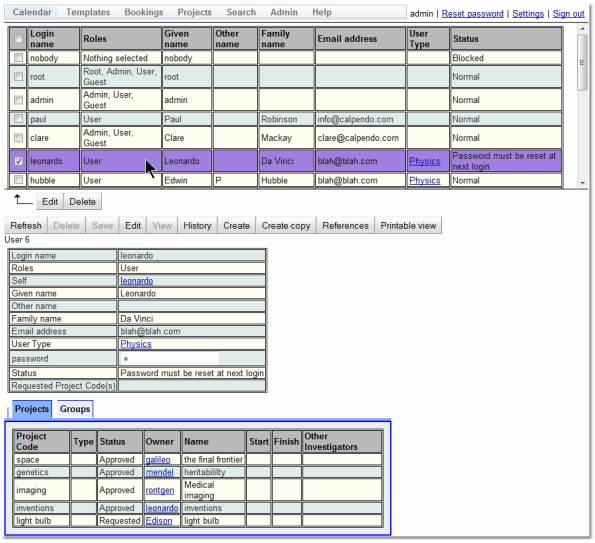
© 2010-2015 Exprodo Software

How To Edit A Single Item

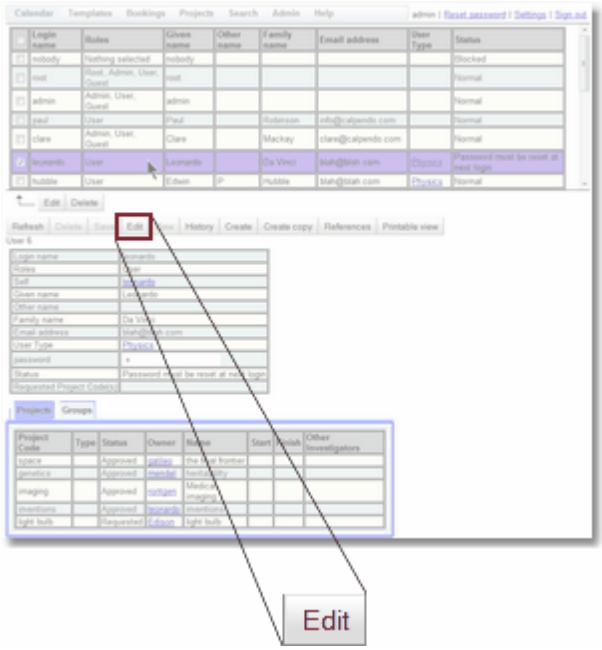
While it is very useful being able to edit multiple items at once, the above mechanism doesn't let you edit every *property*. If the user wants to change the more complex *properties*, such as lists of things, then they need to use a different change mechanism.

For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition 96 chapter.

1. Click anywhere in the row for the item to edit, apart from its check box in the first column. The item's details are shown below the list of items.

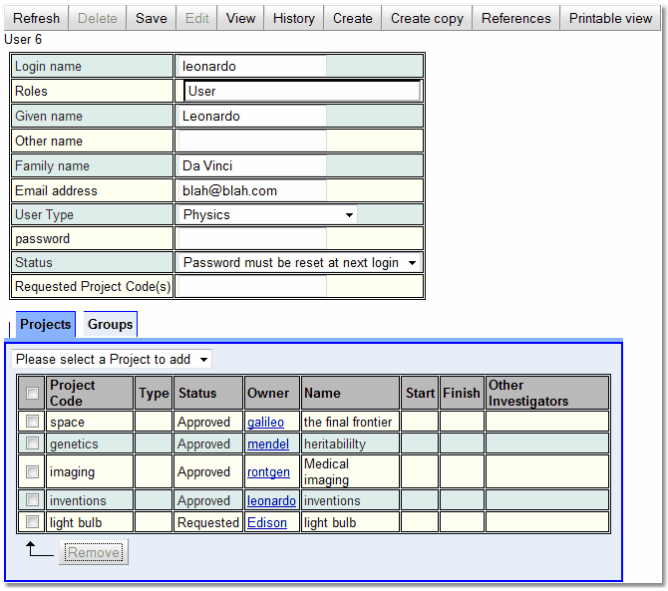

Calendar Templates Bookings Projects Search Admin Help
admin | Reset password | Settings | Sign out
Edit Delete
Refresh Delete Save Edit View History Create Create copy References Printable view
User 6
Projects Groups

2. Press the **Edit** button (not the one in the checked button bar)

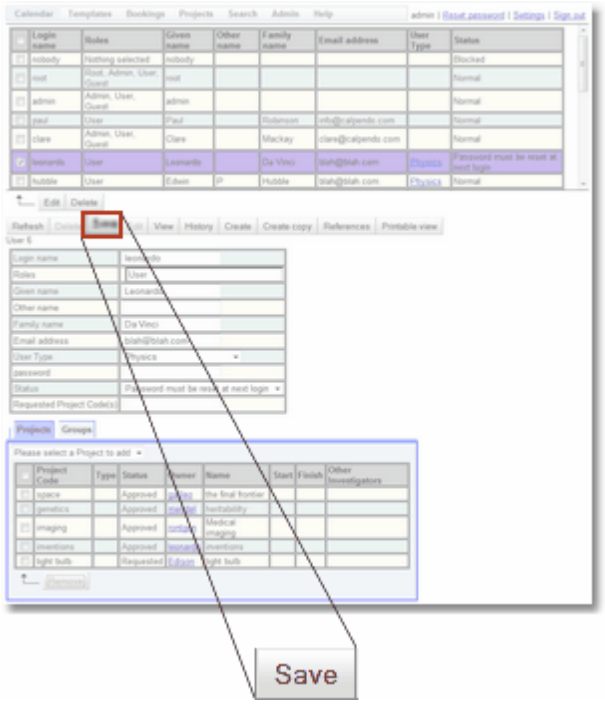


© 2010-2015 Exprodo Software

3. The item is now shown with the *properties* editable. Note that, depending on how *permissions*[448] have been set up, it is possible that some *properties* will still show as read-only.

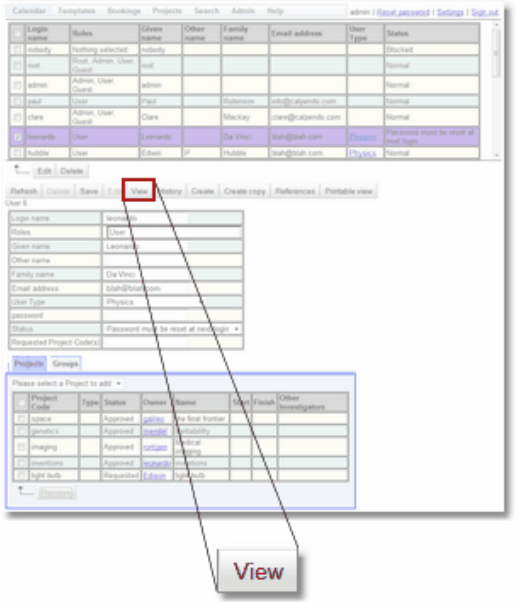


4. Once changing *properties* has been completed, press the **Save** button. The item is then saved and shown read-only again.



© 2010-2015 Exprodo Software

5. To cancel the edit while an item is shown in an editable format, either click on the row for a different item or press the **View** button.



Configuring The Properties Displayed

The user can control which *properties* are displayed in the list of items, in the detailed view of a item. This is done in the Bakery[351].

- Properties Visible In A Biskit List[361] explains how to change the *properties* that are displayed in a list of items. Change the visibility of each of the *properties* defined on an item.
- Properties Visible In Biskit Detail[361] explains how to change the *properties* that are displayed when seeing a detailed view of an item. In this case, it affects both the read-only and editable mode detailed view of an item. Again, change the visibility of each of the *properties* defined on an item.
- Properties Visible In A Collection Editor[361] explains how to change the *properties* that are displayed in collection editors. For example, if there is a *Project*[448] *Biskit Type* that contains a list of **Users**, then the user can change which *properties* are shown to represent those users in the *Project's* list of users.

© 2010-2015 Exprodo Software

3.11 Frequently Asked Questions

If there are any questions relating to the operation of your **Calpendo** at the facility, check the **Frequently Asked Questions** section of **Calpendo**. This can be found on the **Help** menu, and is called *FAQ* 447.

The administrator will configure these *FAQs* so that they are appropriate for the facility (as described in the Configuring Frequently Asked Questions 325 chapter).

3.12 Web Browser Compatability

Calpendo supports the most popular web browsers: Firefox, Safari, Google Chrome and Opera. Internet Explorer is also supported from version 7, but is much slower than the other browsers. Internet Explorer version 6 will mostly work, but will be extremely slow and some behaviour will be incorrect. The users experience with **Calpendo** will be much better if Internet Explorer versions 6 and 7 are avoided, and preferably if Internet Explorer version 8 is avoided as well.

If the user has no choice but to use Internet Explorer versions 6 or 7, then it will help if to install Google ChromeFrame. This is a plug-in for Internet Explorer that allows some web pages to use Google Chrome to render pages.

Name	Version	Comments
Google Chrome	All	Supported
Firefox	3.0 and later	Supported
Apple Safari	4, 5, 6	Supported
Opera	9, 10, 11	Supported
Internet Explorer	6	Not supported. Very slow and doesn't always behave properly.
	7	Supported, but very slow. Use another browser if possible.
	8	Supported, but slower than any of the other major browsers.
	9	Supported, but slightly slower than other major browsers.
	10	Supported

© 2010-2015 Exprodo Software

Part IV

4 Calpendo Administration Guide

The **Calpendo Administration Guide** covers the day-to-day administration that is required for a **Calpendo** installation. This guide should be read in conjunction with the **Calpendo User Guide** [28] which, as well as describing how to make *bookings* [446] and create or edit *projects* [448], also describes how to search for information.

See the **Calpendo Configuration Guide** [198] for information on how to change the properties that exist on a project [206] or the default value of properties on a project [206].

4.1 Booking Administration

When a *booking* [446] is created, it will have a *status* [447] of either *Requested* [449] or *Approved* [446], depending on the Time Templates [450] that are in force for the duration of the *booking*. If there are no applicable *Time Templates* [449] for a newly created *booking*, then its *status* is determined by a *Global Preferences* [447] setting. An administrator will need to approve any *booking* requests and possibly make *bookings* for people who need to make *bookings* outside the normal rules.

The administrator may also need to resolve conflicts between people for which they will need to find out when *bookings* were made or modified. Some *properties* [448] on each *booking* will help with this, but also use the History [126] page.

See Also:

- History [126].
- Configuring Time Templates [215].
- The Bookings [331] section of the Global Preferences [328].

© 2010-2015 Exprodo Software

4.1.1 Booking Properties

The most important *properties*[448] on a *booking*[446] are shown in this table, and these can be seen on the **Bookings Request Page**[139]:

Property Name	Description
Resource	This is the *resource*[449] being booked (the room or equipment etc)
Booker	This is the person that made the booking.
Owner	This is the person that owns the *booking*. When a *booking* is created, if the Global Preferences[331] **Owner Defaults to Project Owner** is set to **True** this is automatically set to be the *project's owner*[448] if there is a *project*, and the *booker*[446] otherwise. Permissions[278] determine whether you can change the *owner* (or indeed any other *property* of a *booking*).
Project	The *project*[448] indicates what the *resource* is being used for. Some *resources* are configured so that their *bookings* do not require a *project*. Some *resources* can be configured so that the *project* must have the *resource* in their *Project Resource Settings*[448] in order to be able to book that *resource*. This enables Booking Rules to be implemented about the maximum number of *bookings* as well as get reports on costs.
Project Resource Settings	This is a special *property* whose value cannot be set by the user. Each *project* has a *Resource Settings property* that is a list of settings for each *resource*. On a *booking*, the *Project Resource Settings* property evaluates to the settings defined on the *booking's project* for the *booking's resource*. Note that if the *project's* settings does not include anything for the *booking's resource*, then the *booking's Project Resource Settings* property will evaluate to null.
Type	This is the *booking type*[447]
Status	A *booking status*[447] is one of **Requested**, **Approved**, **Denied** and **Cancelled**.
Date Range	This indicates both the start and the finish of the *booking*. A **Calpendo** *booking* is currently limited to finishing on the same day that it starts. This may be relaxed in a future version, but *repeat*[449] *bookings* reduce some of the effects of this limitation.
All Day	This records whether the *booking* is for the whole day. All day *bookings* do not require a start and finish time, just a date.
Repeat	This indicates whether and how the *booking* should repeat. Your Permissions[278] may be configured to control who can create or modify a *repeat booking*. See Repeat Bookings[53] in the Creating Bookings[49] page for details of *repeats*.

© 2010-2015 Exprodo Software

Find out when a *booking* was modified and by whom by using the History[126] page. However, a *booking* also records some dates in its *history*[447] to help in this process, and by clicking on a *booking* in the Bookings Calendar[36], it will show these dates. Note that this records the dates but not who performed the actions - use the History[126] page for that.

Property Name	Description
Created	This records when the *booking* was created. This *property* is set automatically and cannot be changed.
Cancelled	This records the last time the *booking* was modified in such a way that its *status* changed and its *status* became **Cancelled**.
Modified	This records when the *booking* was last modified. Again, this is set automatically whenever a *booking* is modified.

Finally, there are some ancillary properties:

Property Name	Description
Description	This is a free-text description of whatever the *booker* wanted to record. This text is displayed with the *booking* on the Bookings Calendar[36].
Duration In Minutes	This is a *property* that is calculated from the *booking's* **Date Range** *property*. This cannot be set directly, but it is useful for reporting.
Warned	This records whether or not the user was shown a warning by a *Time Template*[450] when the *booking* was made or modified.
Template Approved	This records whether the *booking* was approved by a *Time Template*[449], as opposed to being approved manually. This allows you to configure different Permissions[278] and Booking Rules[223] depending on how a *booking* came to be approved.
Reminder information	This stores information about who should be reminded about a *booking* and when. It also records whether the reminder has been sent so that it can't be sent twice.

4.1.2 The Booking Approval Process

The *booking approval process*[446] is very similar to the The User Approval Process[153] and is a matter of changing a *booking's status*[447] from **Requested** to **Approved** if the *booking*[446] is to be allowed, or else to **Denied**. The only major differences are that:

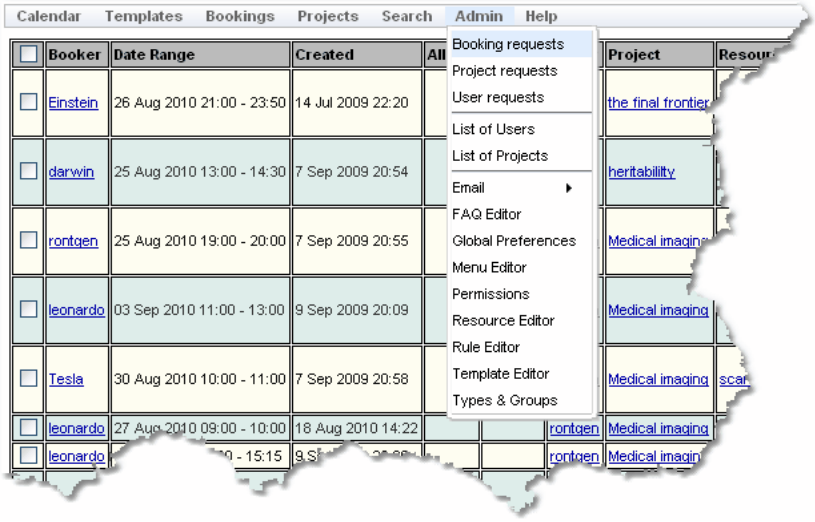
- *Bookings* may be created pre-approved so that no one needs to manually approve any *bookings*, or perhaps only some *bookings*.
- There is no **List of Booking** page because the number of *bookings* would be too large.
- Delete a *booking* request if required, provided Permissions[278] have been suitably configured, however doing so would mean the removal of the *booking* attempt from reports that might be created require.

To approve *bookings*, go to the The Booking Requests Page[139] which shows all the *booking* requests and provides easy ways of approving them. It also offers a way to edit the *bookings*, if required.

© 2010-2015 Exprodo Software

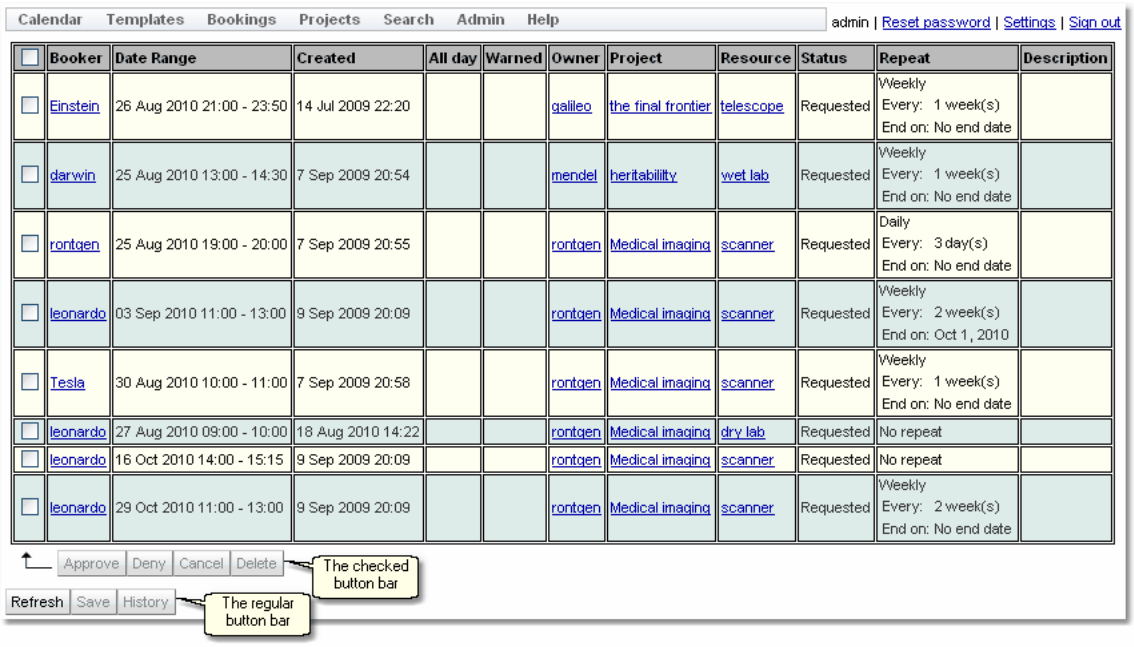
4.1.3 The Booking Requests Page

The **Booking Requests** page shows bookings whose status is **Requested**. By default, the **Booking Requests** page appears on the menu here:



However, the administrator may have configured **Calpendo** so that the menu is different.

This is what the **Booking Requests** page looks like when there are some *bookings*[446] awaiting approval:

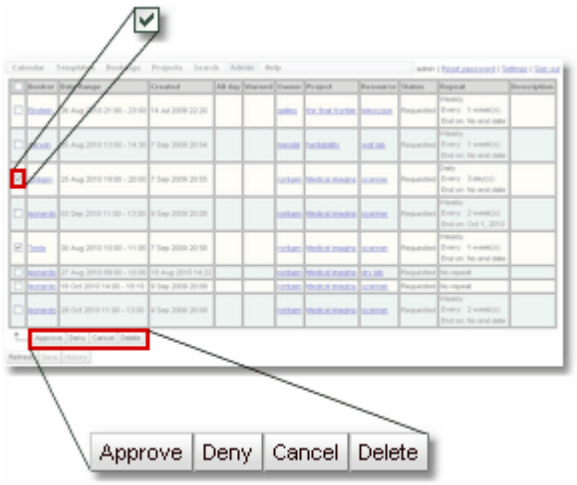


Bookings will only appear in this page if they are repeatable or are currently in the future. Any request which is now in the past will not be viewed.

© 2010-2015 Exprodo Software

Approving Or Denying Bookings

As described in The Booking Approval Process 138, approving or denying *bookings* means changing their *status* 447. The top part of the **Booking Requests** page provides a quick way to do just this. Tick the check box next to each *booking* whose *status* is needed to be changed.



Note that as soon as there are any *bookings* checked, then the checked button bar changes so that it is no longer greyed out.

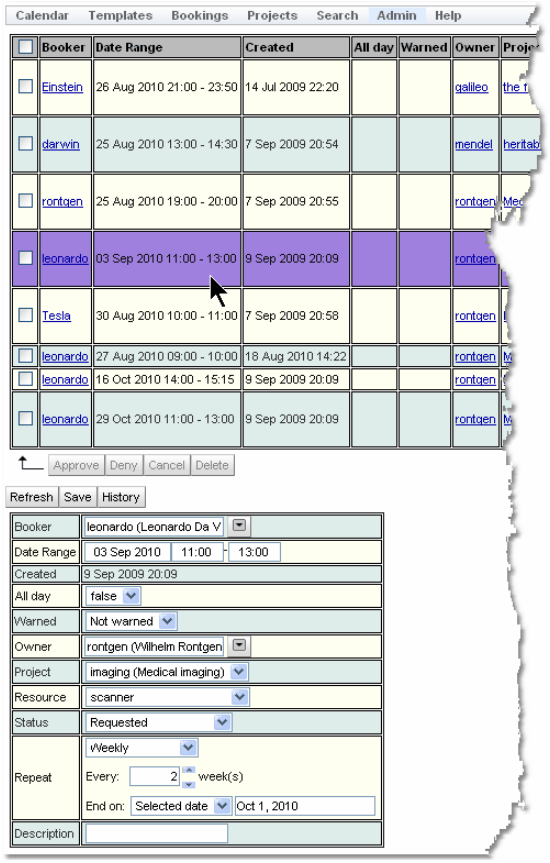
Now press any of these buttons to **Approve**, **Deny**, **Cancel** or **Delete** the ticked *bookings*.

Alternatively, tick the check box in the table's header and every check box will be ticked (or unticked, as appropriate) as a short-cut to ticking all *bookings*.

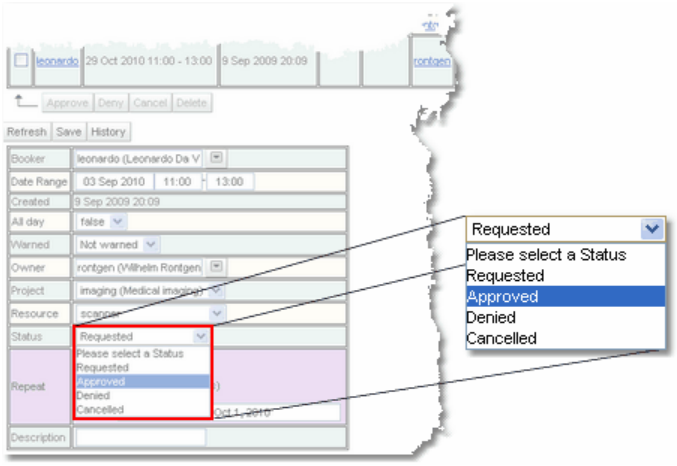
© 2010-2015 Exprodo Software

Changing More Than A Booking's Status

The **Booking Requests** page can be used for more than just changing a *booking's status*. First, click anywhere in the *booking's* row apart from the check box. The *booking's* details will appear at the bottom of the page already in edit mode, and the values can be changed as required.


Calendar Templates Bookings Projects Search Admin Help
Booker Date Range Created All day Warned Owner
Einstein 26 Aug 2010 21:00 - 23:50 14 Jul 2009 22:20 galileo
darwin 25 Aug 2010 13:00 - 14:30 7 Sep 2009 20:54 mendel
rontgen 25 Aug 2010 19:00 - 20:00 7 Sep 2009 20:55 rontgen
leonardo 03 Sep 2010 11:00 - 13:00 9 Sep 2009 20:09 rontgen
Tesla 30 Aug 2010 10:00 - 11:00 7 Sep 2009 20:58 rontgen
leonardo 27 Aug 2010 09:00 - 10:00 18 Aug 2010 14:22 rontgen
leonardo 16 Oct 2010 14:00 - 15:15 9 Sep 2009 20:09 rontgen
leonardo 29 Oct 2010 11:00 - 13:00 9 Sep 2009 20:09 rontgen
Approve Deny Cancel Delete
Refresh Save History
Booker leonardo (Leonardo Da V
Date Range 03 Sep 2010 11:00 13:00
Created 9 Sep 2009 20:09
All day false
Warned Not warned
Owner rontgen (Wilhelm Rontgen)
Project imaging (Medical imaging)
Resource scanner
Status Requested
Repeat Weekly Every: 2 week(s) End on: Selected date Oct 1, 2010
Description

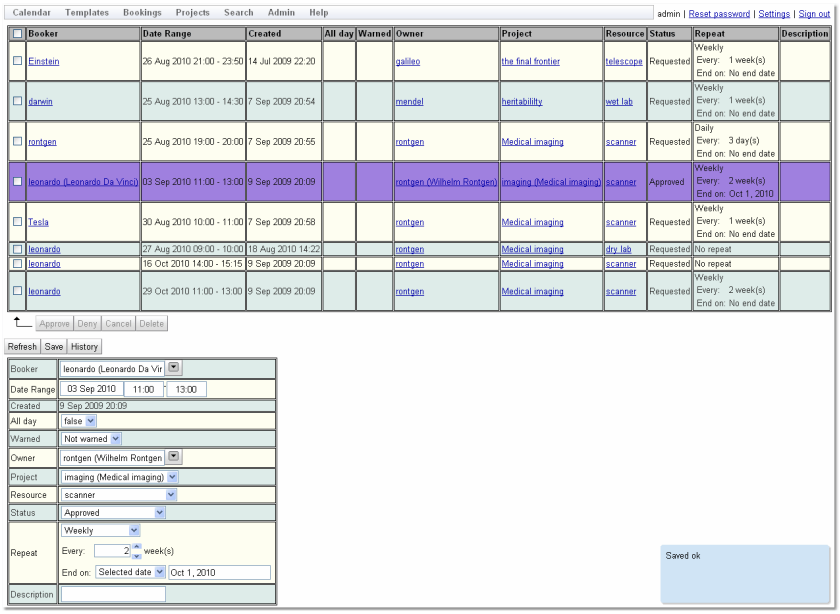
Make changes the *booking* *properties* [448] as required, including changing the status using the status drop-down:



There is no difference between selecting a *status* here and using the buttons in the checked button bar.

© 2010-2015 Exprodo Software

When finished, press the **Save** button. A pop-up will appear briefly in the bottom right corner to indicate the *booking* was saved.


Calendar Templates Bookings Projects Search Admin Help
admin | Reset password | Settings | Sign out
Booker Date Range Created All day Warned Owner Project Resource Status Repeat Description
Einstein 26 Aug 2010 21:00 - 23:50 14 Jul 2009 22:20 galileo the final frontier telescope Requested Weekly Every: 1 week(s) End on: No end date
darwin 25 Aug 2010 13:00 - 14:30 7 Sep 2009 20:54 mendel heritability wet lab Requested Weekly Every: 1 week(s) End on: No end date
rontgen 25 Aug 2010 19:00 - 20:00 7 Sep 2009 20:55 rontgen Medical imaging scanner Requested Daily Every: 3 day(s) End on: No end date
leonardo (Leonardo Da Vinci) 03 Sep 2010 11:00 - 13:00 9 Sep 2009 20:09 rontgen (Wilhelm Rontgen) imaging (Medical imaging) scanner Approved Weekly Every: 2 week(s) End on: Oct 1, 2010
Tesla 30 Aug 2010 10:00 - 11:00 7 Sep 2009 20:58 rontgen Medical imaging scanner Requested Weekly Every: 1 week(s) End on: No end date
leonardo 27 Aug 2010 09:00 - 10:00 18 Aug 2010 14:22 rontgen Medical imaging dry lab Requested No repeat
leonardo 16 Oct 2010 14:00 - 15:15 9 Sep 2009 20:09 rontgen Medical imaging scanner Requested No repeat
leonardo 29 Oct 2010 11:00 - 13:00 9 Sep 2009 20:09 rontgen Medical imaging scanner Requested Weekly Every: 2 week(s) End on: No end date
Approve Deny Cancel Delete
Refresh Save History
Booker leonardo (Leonardo Da Vir
Date Range 03 Sep 2010 11:00 13:00
Created 9 Sep 2009 20:09
All day false
Warned Not warned
Owner rontgen (Wilhelm Rontgen
Project imaging (Medical imaging)
Resource scanner
Status Approved
Repeat Weekly Every: 2 week(s) End on: Selected date Oct 1, 2010
Description
Saved ok

Please note that although we've now changed this *booking's status*, it still appears in the table of *booking* requests. However, the *status* shown in the table has changed. A *booking* will remain in the **Booking Requests** page until the **Refresh** button is pressed, or the user logs out and logs back in again, or refresh's the browser.

© 2010-2015 Exprodo Software

4.2 Project Administration

When somebody first creates a *project*[448] (as described by Creating A New Project[70]), the *project* is created with a *status*[448] of **Requested**. While a *project* has this *status*, nobody can book against it. An administrator must approve the *project* by changing its *status* to **Approved** before it becomes bookable. This and other project-related administration is described in this section.

© 2010-2015 Exprodo Software

4.2.1 Project Properties

A *Project*[448] in **Calpendo** has at least the following *properties*[448]:

Property Name	Description
Project Code	This should be a short, unique code for the *project*. **Calpendo** will make sure that this is not the same as any other *project* (unless it's empty, as would be typical of a new *project* before an administrator assigns a code).
Type	This is the *project's type*[448]. See Types And Groups[265] to understand how types are used. It is possible that **Calpendo** has been configured so that there are no *project types*. If this is the case, then it ought to have been configured so that the *project's type* property is not visible. See Property Visibility for more details.
Status	A *project status*[448] has one of the following values:

Name	Description
Requested	This is the status used when a *project* is first created. *bookings*[446] may not be created for a *project* that is in this state.
Approved	This status means that the *project* is bookable.
Denied	This status indicates that it was decided the *project* should not be allowed, for whatever reason. *Projects* in this state are not bookable.
Terminated	This status indicates that the *project* has been used for making *bookings*, but that the *project* has now come to an end. A terminated *project* is not bookable.
Unbookable	This status indicates the *project* is not bookable for some other reason. Typically, the Project Template[206] is given this status.

Owner	This is the user that owns the *project*. Normally, this would be the user that created the *project*, but an administrator could change the ownership of a *project*.
Name	This is the name of the project, and is intended to be a short description of it
Description	This is a longer description of the *project.*
Project Resource Settings	This is a list of settings, for each *resource*[449], that give per-*project* and per-*resource* information. This is described further below.
Project Service Settings	This is a list of the services available to the project.
Users	This is a list of the users that are associated with the *project*. Only users associated with a *project* may book against that *project*. The exception to this rule is that a user with the **Admin** *role*[450] may book for any *project*.
Project Groups	Which project groups the project is a part of.

© 2010-2015 Exprodo Software

These *properties* represent a reasonable minimum that would be required to get started, but any facility that makes good use of *projects* will be likely to want many more *properties*. **Calpendo** may well have additional *properties* configured for its *projects*. See the Bakery[351] for the details of how to configure *properties* on a *project*.

A *Project Resource Settings* has the following properties:

Property Name	Description
Resource	This is the resource this *Project Resource Settings*[448] pertains to
Number of Sessions	This is the total number of *bookings* that are expected to be used by this *project* for this *resource*.
Minutes Per Session	This is the expected average duration of each *booking* for this project on this *resource*.
Cost Per Hour	This is the cost to be charged for each session.

Again, as for *projects*, your **Calpendo** may have additional *properties* configured for *Project Resource Settings*. Such extra *properties* could be used in *Booking Rules*[446] or in generating reports.

4.2.2 The Project Approval Process

The process of approving new *projects*[448] is highly configurable in **Calpendo** (see Configuring The Project Approval Process[208] for details). This should be configured in a manner that makes sense for your facility. The basic options are whether to have a single-step or a multi-step approval process.

Single-Step Project Approval Process

A single-step approval process is appropriate when there is a single person who decides whether a *project* should go ahead, or the decision is made in a single meeting. Under these circumstances, a *project* changes from its **Requested** state to either **Approved** or **Denied** in a single step. The project's status *property*[448] is sufficient to handle this, and as far as **Calpendo** is concerned, the approval process is merely a matter of somebody (probably an administrator) changing the *project's status*[448].

An administrator has a choice of how to approve *projects* using the single-step approval process. The easiest and preferred option is to use **The Project Requests Page**[146] which shows you just the *projects* whose status is **Requested**. It also lets you modify those *projects* and provides short-cuts to approving or denying new *project* requests. Alternatively, follow the instructions in Modifying Projects[72] and change the *project status* as required, or use the Data Explorer[127] to locate the *projects* to be approved.

There may also be an Automatic Email[297] that tells administrators when there's a *project* that needs approving. If there is, then the email could include a direct HTML link that will take the administrator to a page with the *project* already loaded - making approving it very quick.

© 2010-2015 Exprodo Software

Multi-Step Project Approval Process

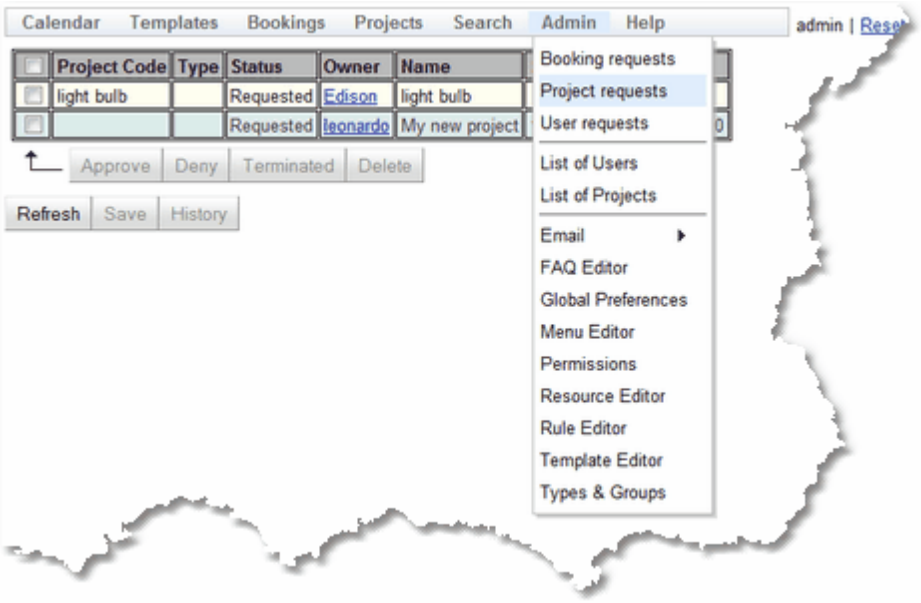
A multi-step approval process is appropriate when there are multiple people who must approve each *project*, and these people make their decisions at different times. To cater for this process, the *project* should be configured with additional *properties* to store the information about whether each person has given their approval. The Permissions[278] can be configured to control who can set these approval *properties*, and even the order in which they must be given. Automatic Emails[297] can be configured to send the next person in the chain an email indicating it's time for them to approve a *project*. The detail of this process is described in Configuring The Project Approval Process[208].

This all means that approving a *project* can mean quite different things, depending on how **Calpendo** has been configured. However, the multi-step *project approval process*[448] can always be characterised as some people modifying *properties* on the *project* while its status is **Requested** and then finally somebody changing the *project's status* to **Approved** or **Denied**. This means the *project approval process* is a matter of all these approvers editing the *project* in turn, and then somebody edits the *project* to change its status.

The Project Requests Page[146] is still the preferred way of achieving each of these changes since this page provides a convenient means of locating the requested *projects*, and the page allows modifications to be made to the *projects* in exactly the same manner as the other methods, as described by Modifying Projects[72] or using Data Explorer[127]. However, the multi-stage approval process might send Automatic Emails with a nested HTML link directly to the *project* in question. If that is done, then locating the requested *project* does not become an issue and so the email could include a link to the **List of Projects** page or Data Explorer[127].

4.2.3 The Project Requests Page

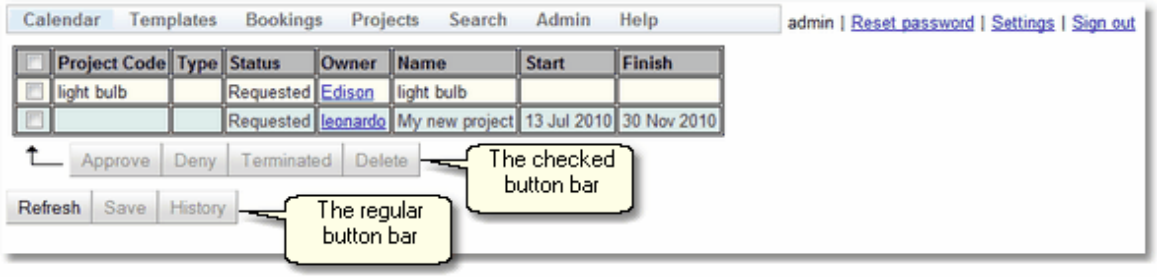
The **Project Requests** page shows you *projects*[448] whose status is **Requested**. By default, the **Project Requests** page appears on the menu here:



However, the administrator may have configured **Calpendo** so that the menu is different.

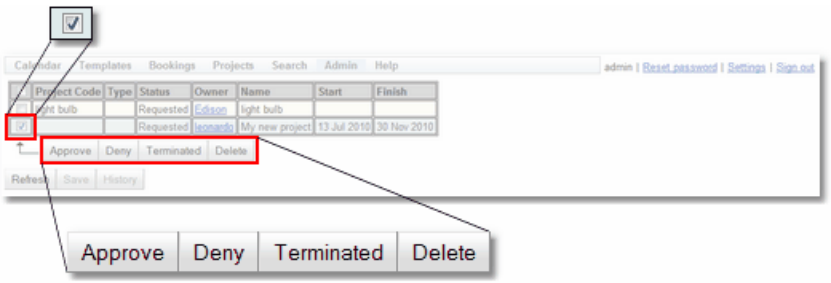
© 2010-2015 Exprodo Software

This is what the **Project Requests** page looks like when there are some *projects* awaiting approval:



Approving or Denying Projects

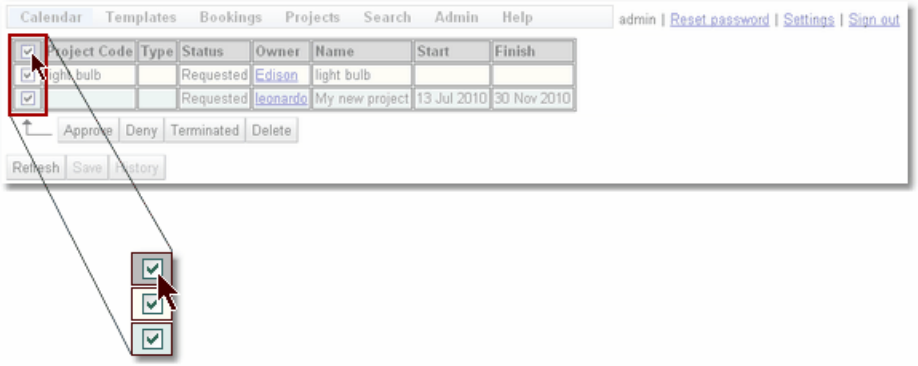
As described in The Project Approval Process [145], approving or denying new *projects* means changing their status, and possibly changing custom *properties* [448] that have been added to a *project*. If using the single step project approval process [145], or needing to perform the last step of the multi-step project approval process [146], then use the top part of the **Project Requests** page to do just this. First, tick the check box next to each *project* whose status requires changing.



Note that as soon as there are any *projects* checked, then the checked button bar changes so that it is no longer greyed out.

Now press any of these buttons to **Approve**, **Deny**, **Terminate** or **Delete** the ticked *projects*.

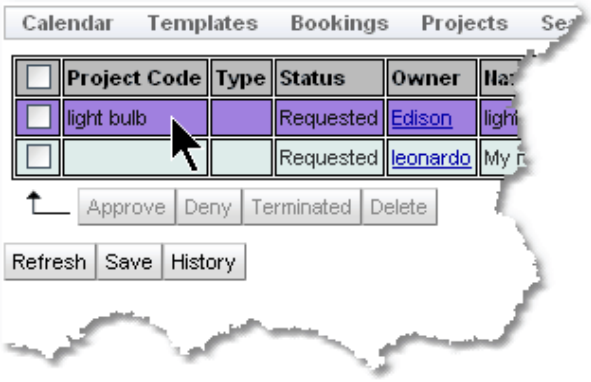
Alternatively, tick the check box in the table's header and every check box will be ticked (or unticked, as appropriate) as a short-cut to ticking all *projects*.



© 2010-2015 Exprodo Software

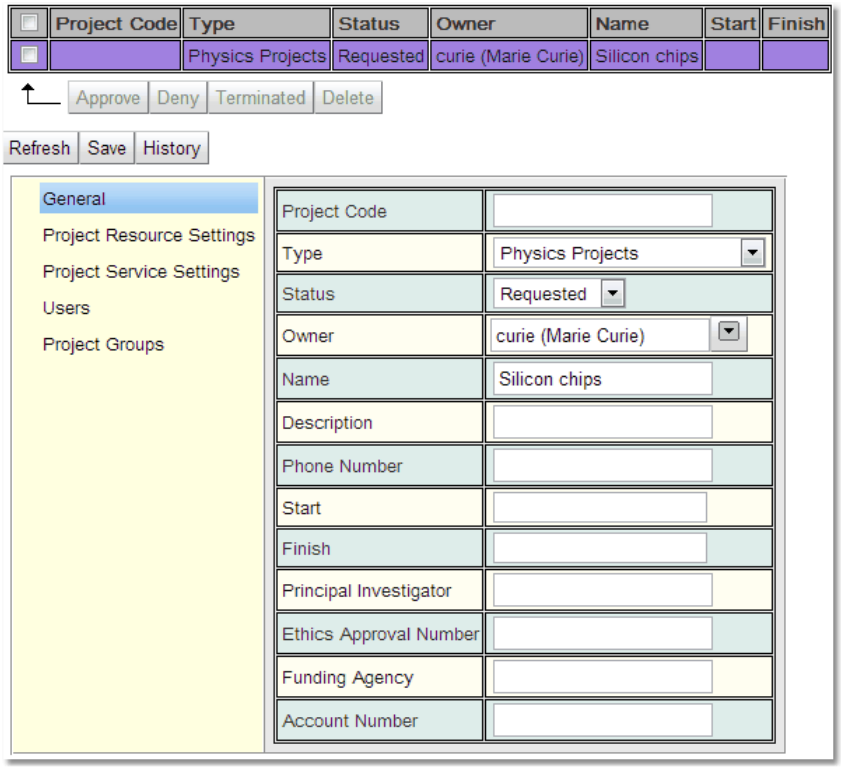
Changing More Than A Project's Status

The **Project Requests** page can be used for more than just changing a *project's status*[448]. First, click anywhere in the *project's* row apart from the check box.



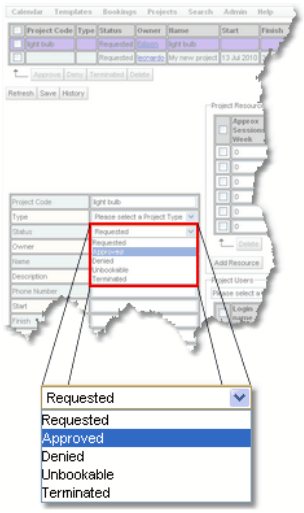
Selecting a single project to edit or view its details

The *project's* details will appear at the bottom of the page already in edit mode, and you can change the values as required. The *projects* may have been configured with different *properties* from those shown in this picture, so may look different.

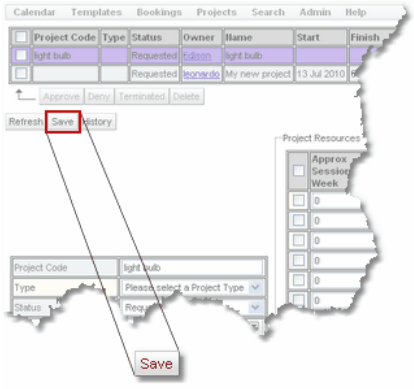


© 2010-2015 Exprodo Software

The *project property* values can be changed as required, including changing the status using the status drop-down:

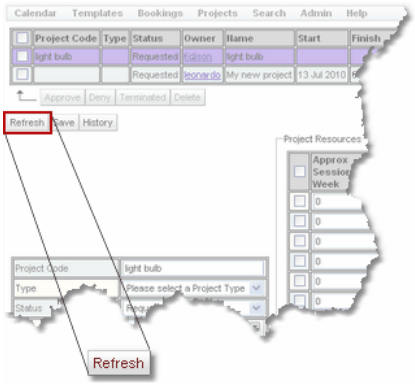


When finished, press the **Save** button.



A pop-up will appear briefly in the bottom right corner to indicate the *project* was saved.

Please notice that although the *project's status* has now changed, it still appears in the table of *project* requests. However, the status shown in the table has changed. A *project* will remain in the **Project Requests** page until the **Refresh** button is pressed, or the user logs out and logs back in again, or refreshes the browser.



© 2010-2015 Exprodo Software

4.3 User Administration

When somebody first registers with **Calpendo** (as described by Getting A User Account 28), a new user is created with a status of **Requested**. While a user has this status, they are unable to log in. An administrator must then approve the user by changing their status to **Approved**. This and other user-related administration is described in this section.

4.3.1 User Properties

This chapter describes the properties of a user and what they are for, but not how to change them. See Modifying Users 163 for a description of how to change the *properties* 448 on a user. The licence for **Calpendo** counts the users with status **Normal** (whose accounts have not expired) and **Password must be reset at next login**.

Identity

Every user has a combination of authentication method and login name which must be unique. The login name may contain letters, numbers, hyphen, underscore or full stop. A login name may be changed as long as the new name adheres to these constraints.

Roles

There can be up to 32 *roles* 450 defined in the system, including the predefined and special **Root** and **Admin** *roles*. A user with the **Root** role always has *permission* to do anything in the system. A user with the **Admin** role is allowed to perform administrative tasks. However, **Calpendo** can be configured so that the **Admin** role is not required for administrative tasks. The remaining 30 possible roles can be configured as required, and they are normally associated with setting up *Permissions* as required for your facility. See User Roles 368 in the Bakery 351 section of the **Calpendo Configuration Guide** for how add new *roles*.

Users with the **Root** role have the following special properties:

- *Permissions* are not checked. *Permission* is always granted.
- The *booker* 446 of a *booking* 446 can be changed. Normally, the *booker* is always the user that first created the *booking* and only users with the **Root** role can break that.
- They are allowed to log in, even while the licence has expired. This allows you to recover such situations once you have a new licence.
- They can change the *status* 450 of a user to make them active (for example, by changing the status from **Blocked** to **Normal**), even if it means there will be more active users than the licence allows. However, this may not be a good thing to do since only **Root** would then be able to use the system.
- Normally, a user can only allocate other users with roles that they already have. However, if a user has the **Root** role, they can add any role to another user.

© 2010-2015 Exprodo Software

Users with the **Admin** role have the following special properties:

- By default, *Permissions* are created that allow a user with the **Admin** role to **create**, **update** and **delete** almost all *Biskit Types*[446]. They can also update the database schema when using the Bakery[351]. However, these *Permissions* are changeable and may be added, removed or modified for any user by a user with **Admin** privileges.
- *Time Templates*[449] may not apply to users with the **Admin** role, depending on the *global preferences* setting Templates -> Templates Apply To Admin[349]
- *Booking Rules* may not apply to users with the **Admin** role, depending on the *global preferences* setting Booking Rules -> Booking Rules Apply To Admin[345]
- When creating a *booking*, the default status assigned is specified by the *global preferences* setting Bookings -> Admin Default Booking Status[331] whereas for regular users it is decided by Bookings -> User Default Booking Status[331].
- They can create *bookings* against any *project*, whereas regular users can only book against *projects* whose status is **Approved** and *projects* for which they appear in the *project's* users selection.
- When the **Calpendo** licence has fewer than 30 days remaining, users with the **Admin** role will receive a warning each time they log in. Regular users only receive a warning in the 7 days prior to expiration. Note that 30 days' grace period is allowed, so that there is always time after expiration before the licence must be renewed.
- When a user with the **Admin** role logs in for the very first time, they are assigned the menu that has been configured for **Admin** users in Global Preferences -> Menus -> Default Admin Menu[341]. A user that has neither the **Root** nor the **Admin** role will be assigned the menu specified in Menus -> Default User Menu[341] when they first log in.

The final difference that users with the **Root** or **Admin** role may notice is that each *Biskit Type* can be configured in the Bakery[351] to be visible to users with the **Root** role, users with the **Admin** role, everybody or nobody. For example this provides a way to reduce the number of *bisikit types* that users may search for, for example.

Names And Email

A user's name is split into three parts: their given name (usually their first name), the family name (usually their last name) and their other name (for any middle names). **Calpendo** uses these names for display and in reports, but not for anything else. For example, it is not a requirement that a user's name is unique.

A user's email address is important because **Calpendo** sends emails for various reasons (Automatic Emails[297] and Manual Emails[172]).

Type

Calpendo may be configured so that users are asked for a *type*[450] when they first register. The *type* is a means of segregating users into non-overlapping sets, which may be used by the facility for assigning *Permissions*[448], *Booking Rules*[446], and *Time Templates*[450]. The possible values for a *user type* are configured in Configuring Types And Groups[266].

© 2010-2015 Exprodo Software

Password

The users password is stored here, but will only be shown as a number of dots for security reasons. If there is no password the box will be empty.

Status

A *user's status*[450] indicates whether they can log in. It can take the following values:

Status	Description
Requested	This is the status given to a user when they first register. A user whose status is **Requested** cannot log in.
Normal	This is the normal status for a user who can log in.
Password must be reset at next login	A user whose status is **Password must be reset at next login** can log in, but they will be forced to change their password as soon as they do so. If an administrator needs to reset a user's password, then it's useful also to set their status to this value at the same time.
Blocked	A user whose status is **Blocked** cannot log in. Typically, use this status for a user that needs to be stopped from logging in for some reason.
Denied	A user whose status is **Denied** cannot log in. Typically, use this status when not approving a new user request, although there is also have the option of deleting the user.
Expired	A user whose status is **Expired** cannot log in. Either their expiry date has passed or the administrator has decided they no longer need to use the system.
Lurker	A user whose status is Lurker cannot login but will receive automatic and booking reminder e-mails.

The reason that both **Blocked** and **Denied** statuses exist is so that an Automatic Email[297] can be sent to the affected user with a message that is suitable for the situation.

Expiry Date

This allows the administrator to set an expiry date for the users login. When this field is edited a calendar will appear, just select the day the login is to expire on.

Last Login and Last Login From

This records the last date and time the user logged in and the IP address of the machine they logged in from.

Groups

A user may belong to any number of *user groups*[450], and group membership can be used for assigning *Permissions*, *Booking Rules* and *Time Templates*. The *groups* that exist are configured in Configuring Types And Groups[266].

© 2010-2015 Exprodo Software

4.3.2 The User Approval Process

When users first register, their *status*[450] is set to **Requested**. While a *user's status* is **Requested**, they will not be able to log in. Approving new users, and thereby allowing them to log in, simply means changing their *status* to **Normal**. Deny the user request by changing their *status* to **Denied** or delete the user. It is also useful to define the users *roles*[450] at this point. A new user can be assigned any *role* the person doing the assigning has. User approval can be done in the following places:

- **The User Requests**[153] page[153] shows just the users whose status is **Requested**. It also provides short-cuts to approving or denying user requests and allows for user detail modification.

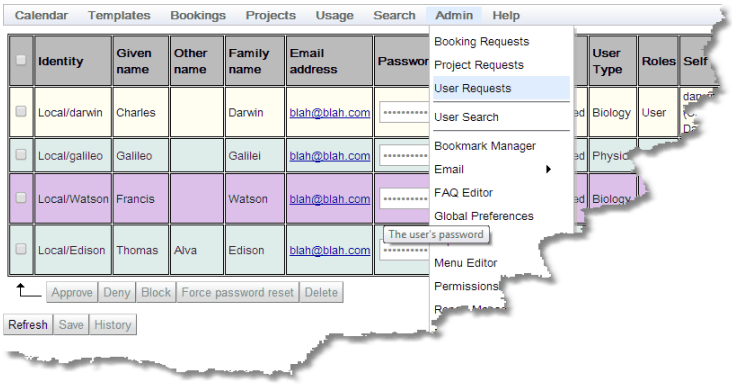
- Modifying Users[163] describes how to use the other options for modifying users. That is, the **User Search** page and the Data Explorer[127].

By default, **Calpendo** has an Automatic Email configured that will tell all administrators when there is a new user request. The email received contains links to approve or deny the new user request. Just click on the links in the email to approve or deny new users.

Please note that if a *resource*[449] requires a *project*[448] to be entered when a *booking*[446] is made, a user cannot book that *resource* unless they are assigned to the *project* being used to make the *booking*.

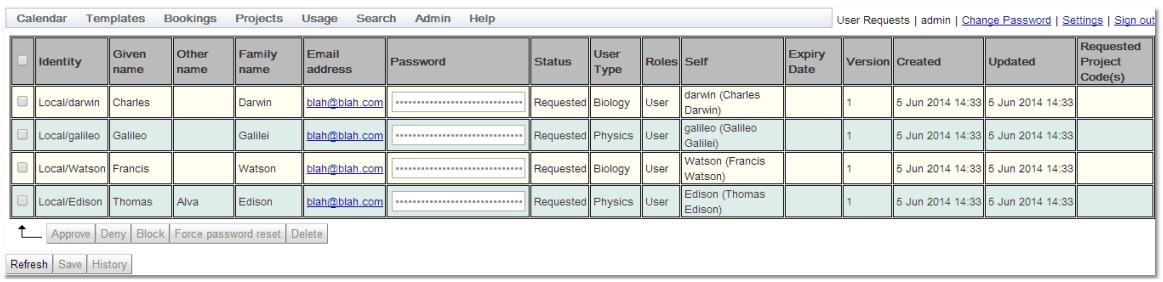
4.3.3 The User Requests Page

The **User Requests** page shows users whose status is **Requested**. By default, the **User Requests** page appears on the menu here:



However, your administrator may have configured the menus differently.

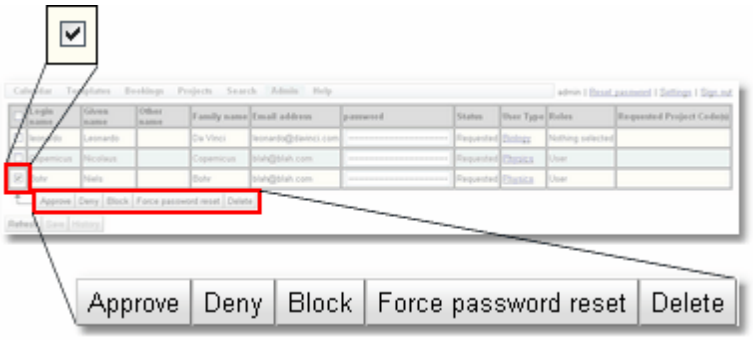
This is what the **User Requests** page looks like when some users are awaiting approval:



© 2010-2015 Exprodo Software

Approving Or Denying Users

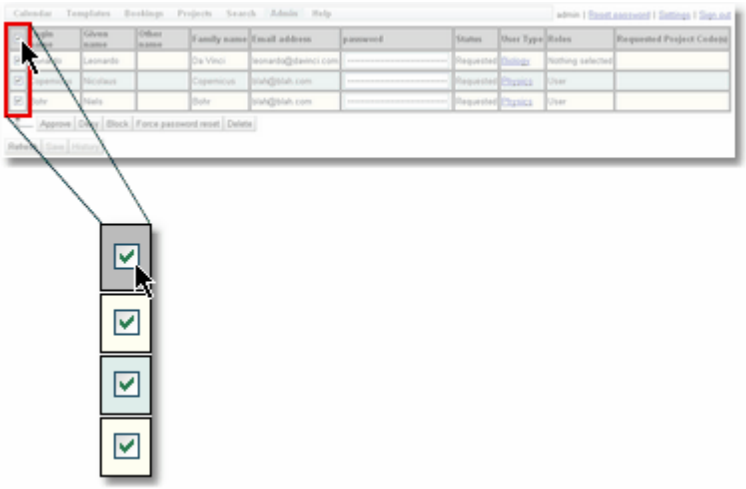
As described in The User Approval Process 153, approving or denying new users means changing their *status* 450. The top part of the **User Requests** page provides a quick way to do just this. Tick the check box next to each user whose *status* to be changed.



Note that as soon as there are any users checked, then the checked button bar changes so that it is no longer greyed out.

Now press any of these buttons to **Approve**, **Deny**, **Block** or **Delete** the ticked users. Press **Force password reset** to allow the users to log in, but force them to change their password.

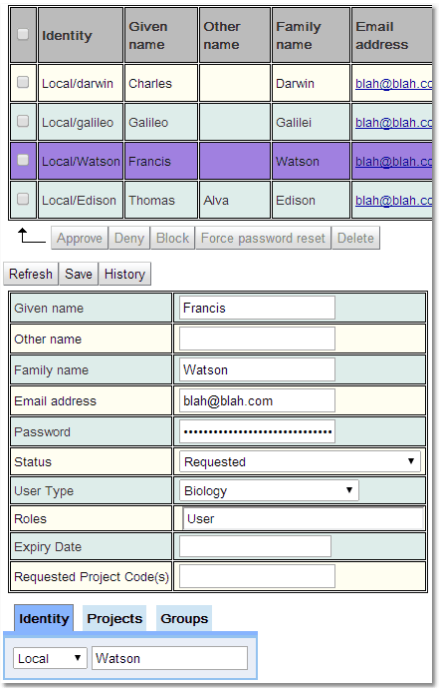
Alternatively, tick the check box in the table's header and every check box will be ticked (or unticked, as appropriate) as a short-cut to ticking all users.



© 2010-2015 Exprodo Software

Changing More Than A User's Status

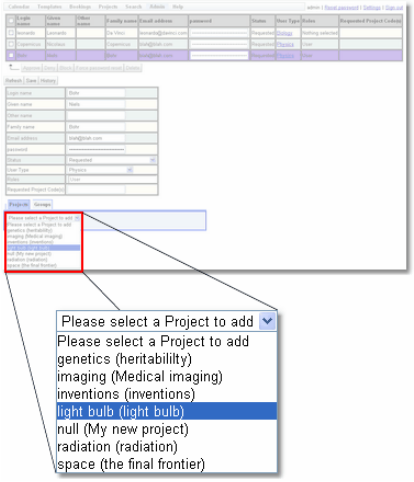
The **User Requests** page can be used for more than just changing a *user's status*. First, click anywhere in the user's row apart from the check box. The user's details will appear at the bottom of the page already in edit mode, and the values can be changed as required. It is useful to define the *users roles* [450] at this point. A new user can be assigned any *role* the person doing the assigning has. If the assignee has **Admin**, **Guest** then they can assign **Admin** and **Guest** *roles* to another user.



Choose the *status* you want the user to have from the drop-down.

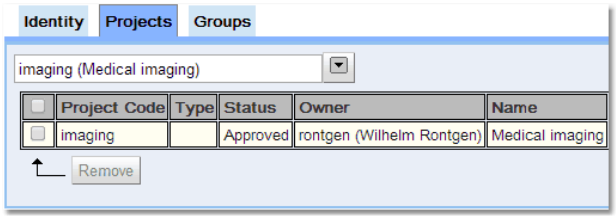
There is no difference between selecting a *status* here and using the buttons in the checked button bar.

To associate this user with some *projects* [448], select the **Projects** tab at the bottom and select the *projects* you want to add.

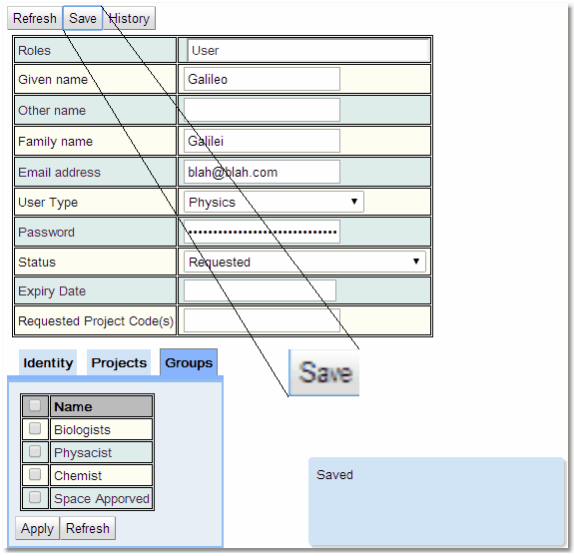


© 2010-2015 Exprodo Software

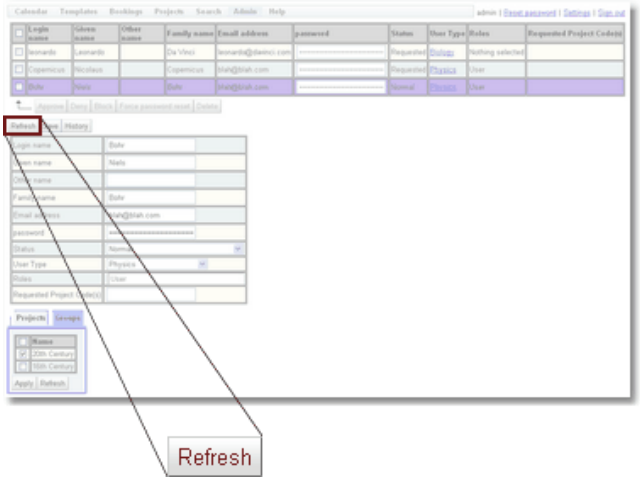
After *projects* have been added, they are shown in the table.



When finished, press the ***Save*** button. A pop-up will appear briefly in the bottom right corner to indicate the user was saved.



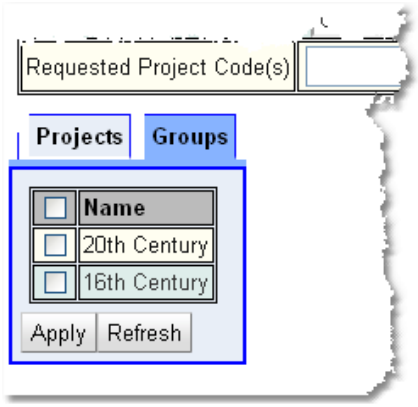
Although we have now changed this *user's status*, it still appears in the table of user requests. However, the *status* shown in the table has changed. A user will remain in the **User Requests** page until the **Refresh** button is pressed, or the browser is refreshed.



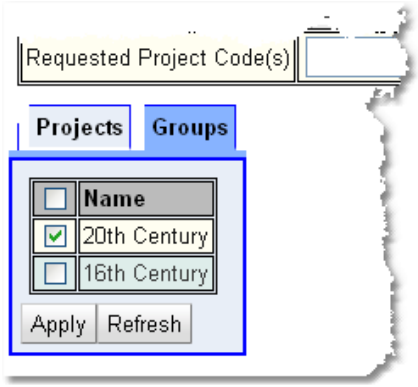
© 2010-2015 Exprodo Software

Changing The User's Group Membership

To associate the user with some *users groups*[450], first select the user's row, as shown above in Changing More Than A User's Status[155], and then click the **Groups** tab.



The check boxes indicate what *groups* this user is a member of. New users will not be a member of any *groups*, so no boxes are checked. Tick the check boxes next to the *groups* this user needs to be a member of.



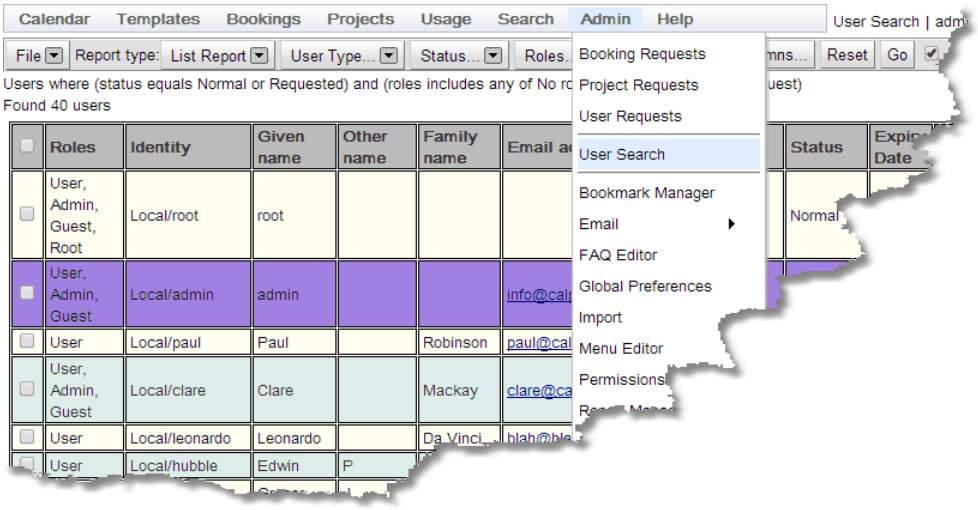
Once all the required *user groups* are chosen, press the **Apply** button. Once done, a pop-up showing the number of user groups that have been changed will appear.

The user does not need to save the user information if only the *group* memberships have changed. That's because *group* memberships are a *property*[448] of the **User Group** and not the **User**, and it's also why there's a separate **Apply** button to save the *group* membership changes.

© 2010-2015 Exprodo Software

4.3.4 User Search

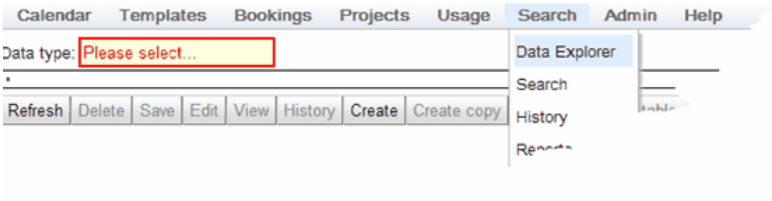
To search for users use the dedicated **User Search** page, which by default appears on the menu here:



The administrator may have configured your **Calpendo** so that the menu is different.

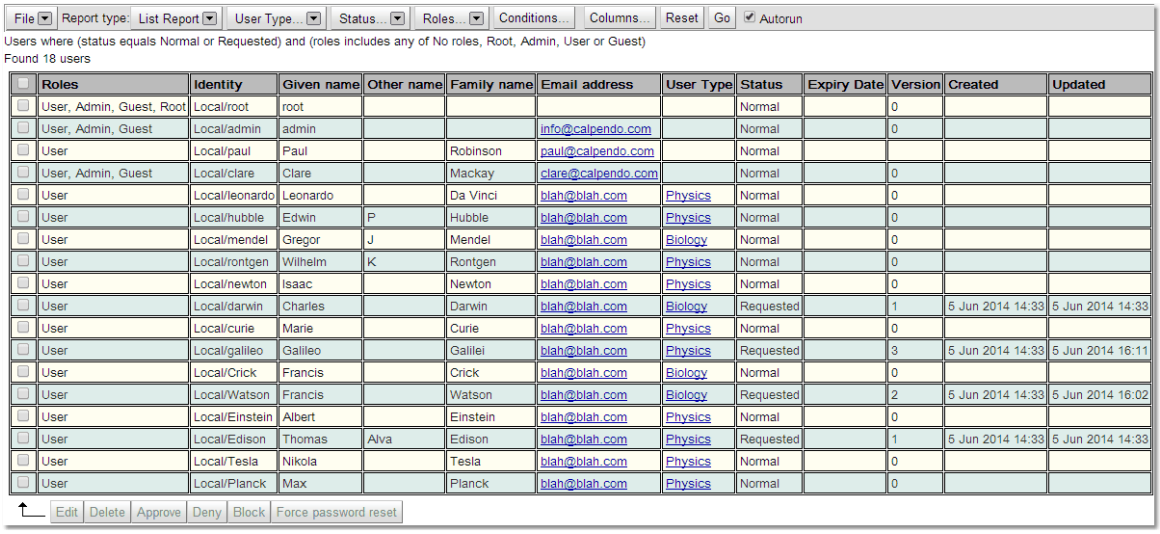
The alternative way to get a list of users is by using the Data Explorer[127] and choosing **User** as the *Biskit Type*[446]. It doesn't make any difference which method is chosen - these pages behave almost identically apart from **User Search** being dedicated to users, while Data Explorer[127] can handle any *Biskit Type*.

By default, Data Explorer[127] appears on the menu here:



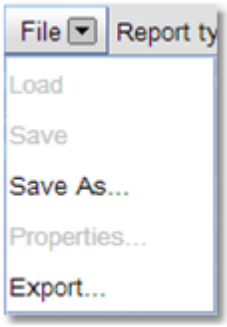
© 2010-2015 Exprodo Software

This is what the **User Search** page looks like when it is first displayed:



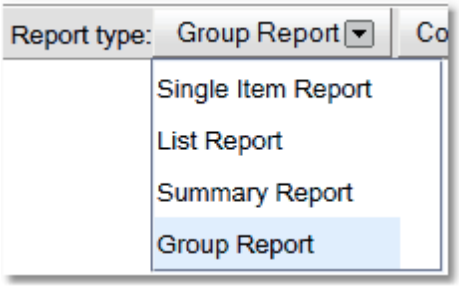
Roles	Identity	Given name	Other name	Family name	Email address	User Type	Status	Expiry Date	Version	Created	Updated
User, Admin, Guest, Root	Local/root	root					Normal		0		
User, Admin, Guest	Local/admin	admin			info@calpendo.com		Normal		0		
User	Local/paul	Paul		Robinson	paul@calpendo.com		Normal				
User, Admin, Guest	Local/clare	Clare		Mackay	clare@calpendo.com		Normal		0		
User	Local/leonardo	Leonardo		Da Vinci	blah@blah.com	Physics	Normal		0		
User	Local/hubble	Edwin	P	Hubble	blah@blah.com	Physics	Normal		0		
User	Local/mendel	Gregor	J	Mendel	blah@blah.com	Biology	Normal		0		
User	Local/rontgen	Wilhelm	K	Rontgen	blah@blah.com	Physics	Normal		0		
User	Local/newton	Isaac		Newton	blah@blah.com	Physics	Normal		0		
User	Local/darwin	Charles		Darwin	blah@blah.com	Biology	Requested		1	5 Jun 2014 14:33	5 Jun 2014 14:33
User	Local/curie	Marie		Curie	blah@blah.com	Physics	Normal		0		
User	Local/galileo	Galileo		Galilei	blah@blah.com	Physics	Requested		3	5 Jun 2014 14:33	5 Jun 2014 16:11
User	Local/Crick	Francis		Crick	blah@blah.com	Biology	Normal		0		
User	Local/Watson	Francis		Watson	blah@blah.com	Biology	Requested		2	5 Jun 2014 14:33	5 Jun 2014 16:02
User	Local/Einstein	Albert		Einstein	blah@blah.com	Physics	Normal		0		
User	Local/Edison	Thomas	Alva	Edison	blah@blah.com	Physics	Requested		1	5 Jun 2014 14:33	5 Jun 2014 14:33
User	Local/Tesla	Nikola		Tesla	blah@blah.com	Physics	Normal		0		
User	Local/Planck	Max		Planck	blah@blah.com	Physics	Normal		0		

The following section goes through the options available after the search has been completed, enabling the search to be changed, saved or the data exported, as well as change the information viewed by the search. For more information on any of the features shown read the Search[108] chapter.



Save As: Allows saving of the report for later use.

Export: Allows export of the information found in the search to a file.



This button determines what type of report will be seen:

Single Item Report[449] Displays a single record that matches the search.

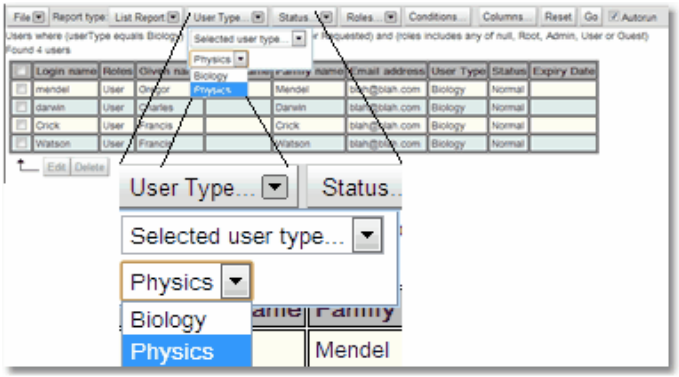
List Report[447]: Lists each record found that matches the search one under the other.

Summary Report[449]: Allows the information found to be summarised.

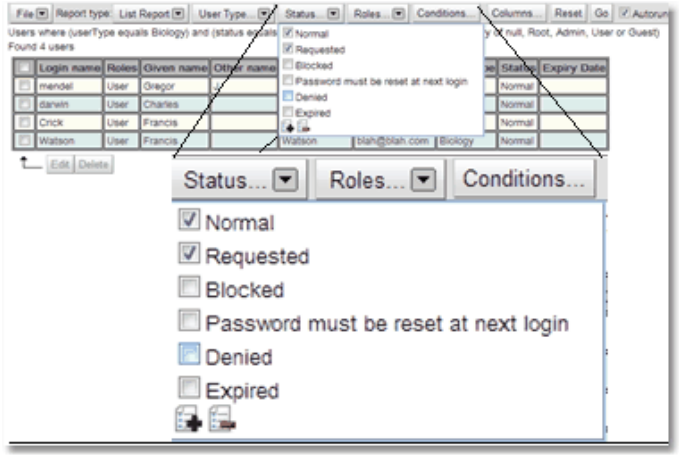
Group Report[447]: Allows the report to be shown by groupings with a count of the number of records that fit each group.

© 2010-2015 Exprodo Software

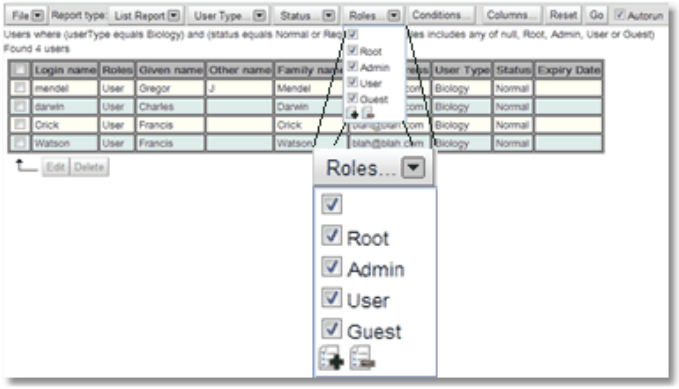
To change the search parameters



Choose **Selected User Type** then select a *user type*[450] from the drop down options.



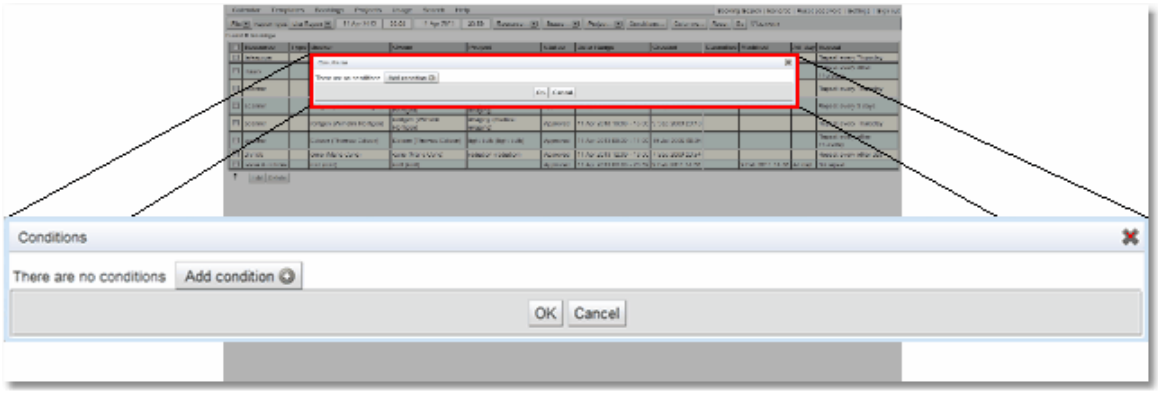
Then select the *user status*[450] required.

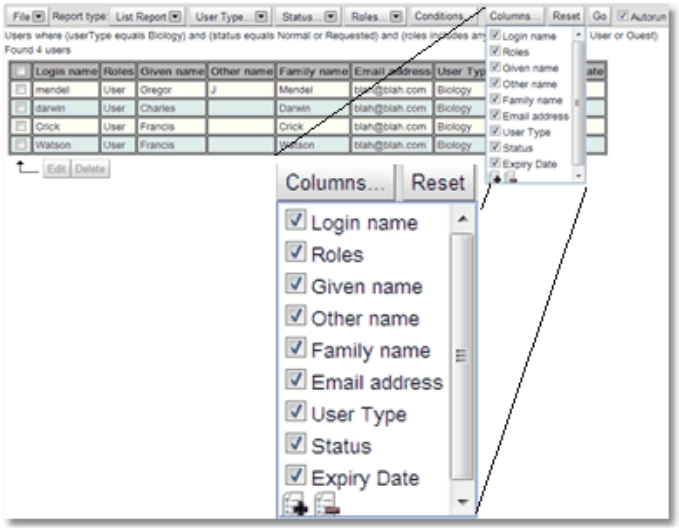


And select the *user roles*[450].

© 2010-2015 Exprodo Software

Set up additional *conditions*[447] for the search. For more information on how to set up *conditions* read the information in the Search[108] page.





And change the columns displayed in the output.

Reset: Resets the search *conditions* to the defaults set up on starting the search.

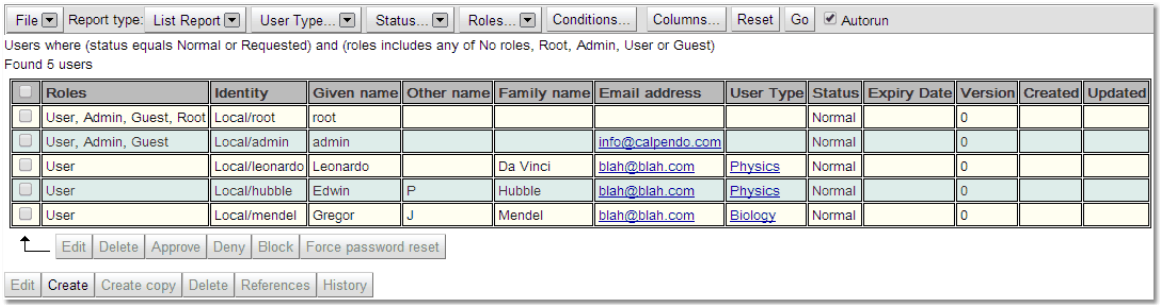
Go: Runs the search with the current *conditions*

Autorun: If this button is ticked then searches will run as soon as any *conditions* change. If the user will be changing a number of the *conditions* of the search, or searches will return large amounts of information it is more efficient if this is not ticked.

© 2010-2015 Exprodo Software

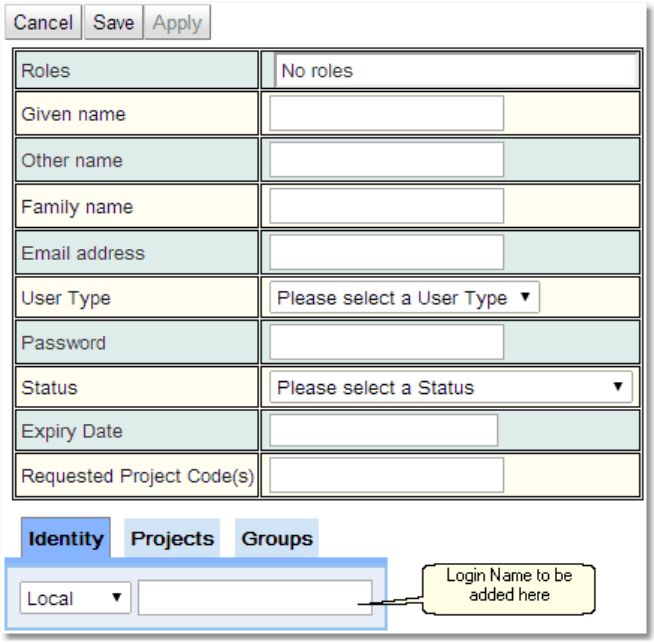
4.3.5 Creating Users

To add a new user that has not been requested go to the **User Search** page. There will be a list of the current users and underneath a menu bar with **Create** button enabled.



For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition 96 chapter.

In order to understand more about the *properties* 448 used for a user read the section on User Properties 150. The login name to be used will be added under identity after choosing the appropriate authentication method. Also in order to understand more about setting up *projects* 448 and *groups* 450 for a user read the section on The User Requests Page 153 and Modifying Users 163.



© 2010-2015 Exprodo Software

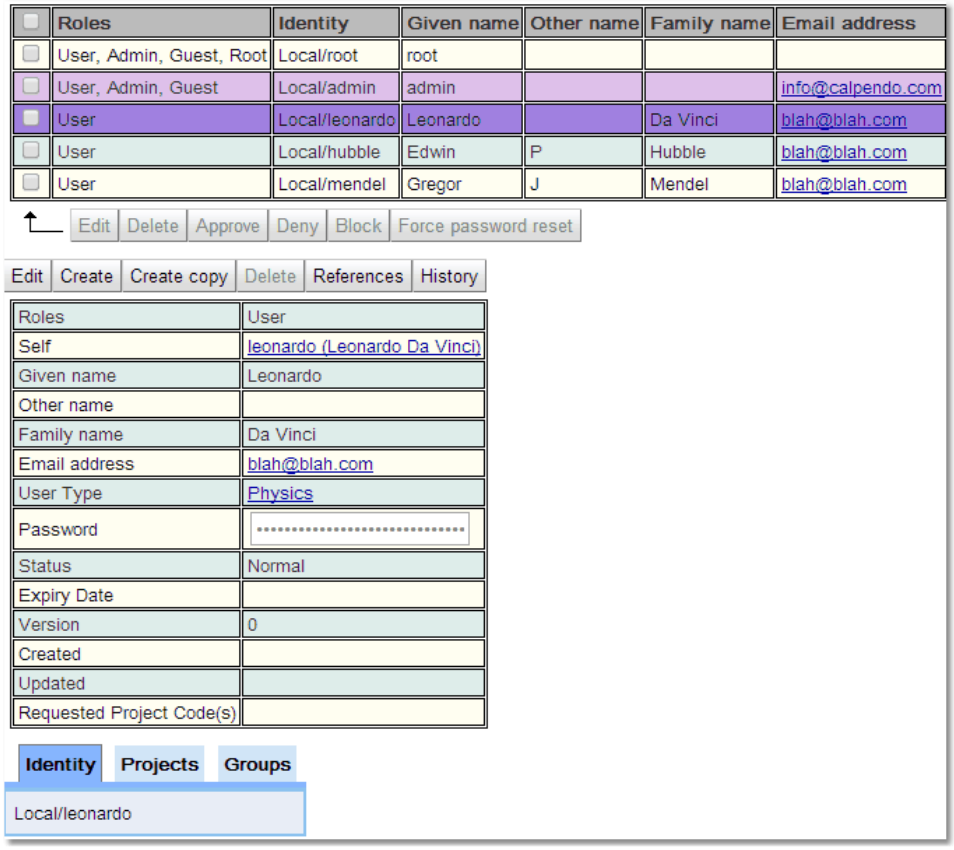
4.3.6 Modifying Users

To see, change or delete users, go to a list of users. Either use the dedicated **User Search** page, or use Data Explorer[127] as explained in the User Search[158] chapter to get the list.

Once the list of users has been returned they can be modified. Depending on which page was used to get the list of users either go to the Data Explorer[127] or Search[108] pages for more details on editing and saving. Described below is how to specifically change a *user's project associations*[450] and *user groups*[450].

Changing A User's Password

Once the list of users is displayed, click on the user whose password needs to be changed.

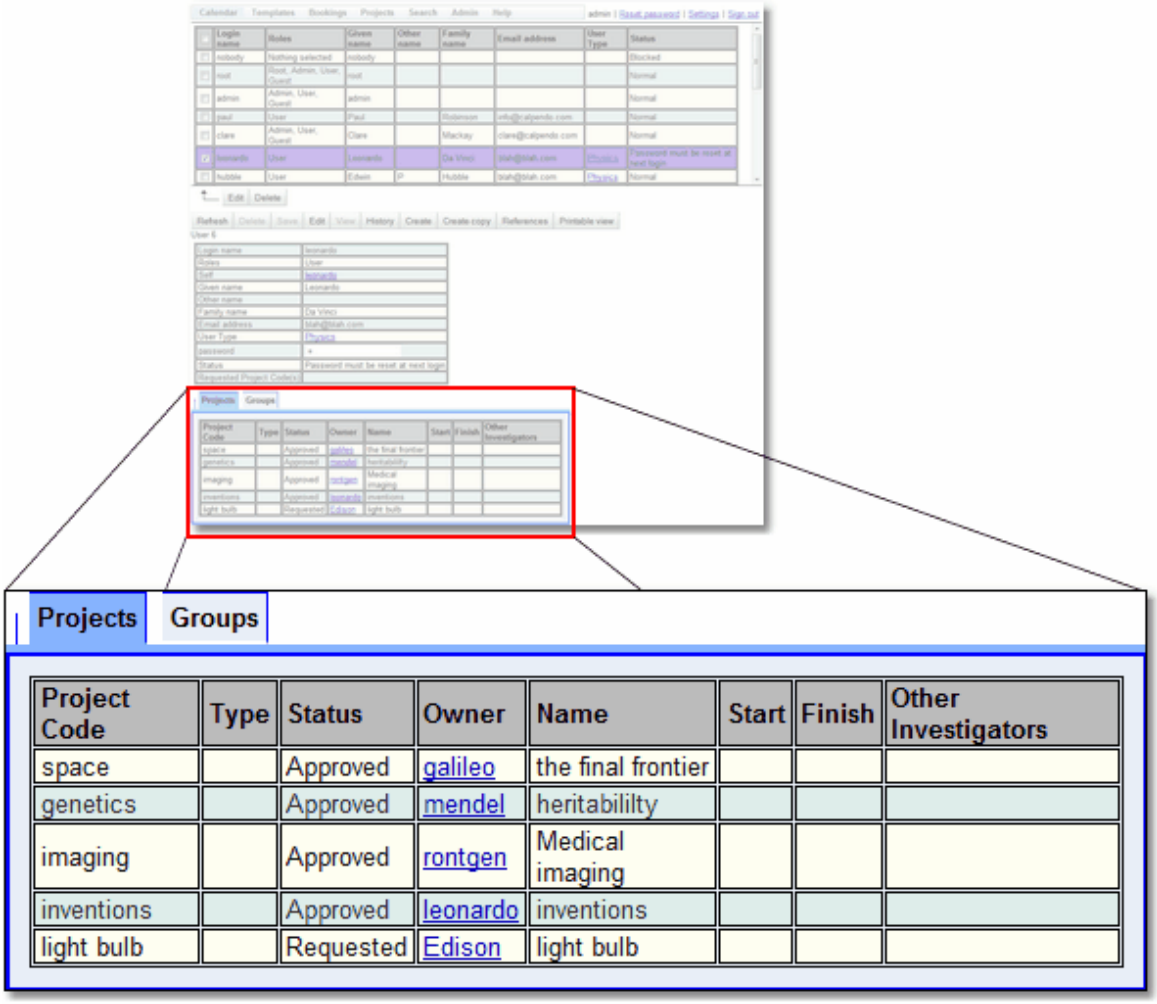


Click on the **Edit** button, once in edit mode any of the users details may be changed. If changing a users password it may be an idea to change the **Status** to **Password must be reset at next login**, in order that the user must change their password again when they log in. Once the details are changed click **Save** (this will have replaced the **Edit** button).

© 2010-2015 Exprodo Software

Changing A User's Project Associations

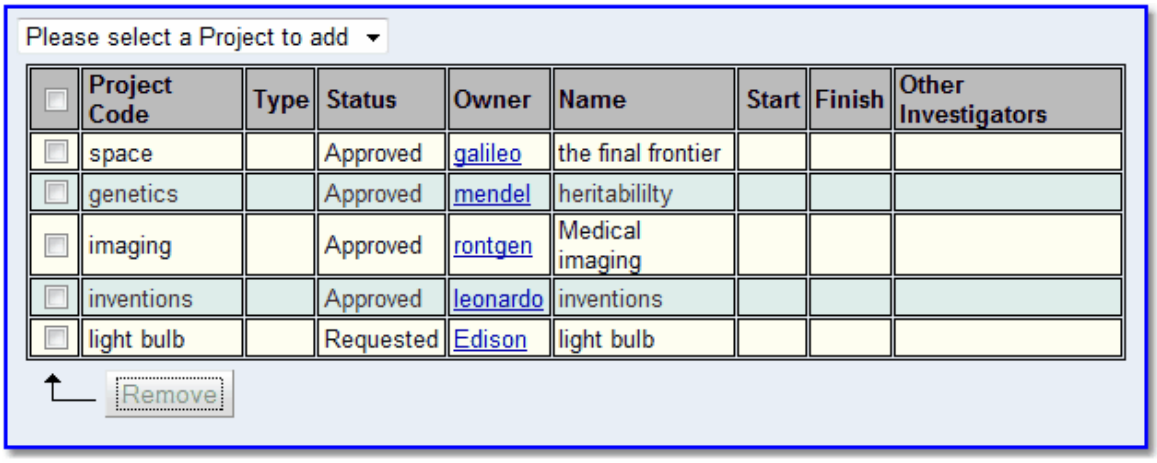
With a user's detail showing, as described above, the *user's project associations* are shown in the **Projects** tab at the bottom of the page.



Projects | Groups

Project Code	Type	Status	Owner	Name	Start	Finish	Other Investigators
space		Approved	galileo	the final frontier			
genetics		Approved	mendel	heritabililty			
imaging		Approved	rontgen	Medical imaging			
inventions		Approved	leonardo	inventions			
light bulb		Requested	Edison	light bulb			

In edit mode, the ***Projects*** tab changes to this:



© 2010-2015 Exprodo Software

This has its own checked button bar:

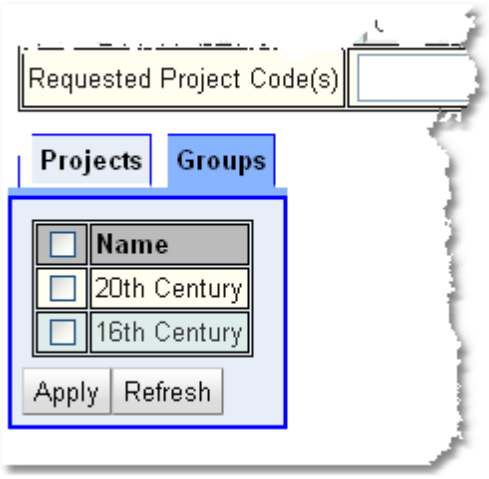
1. Tick the check boxes of any *project*[448] to be removed.
2. The **Remove** button will change so that it is not greyed out once a check box is ticked.
3. Press the **Remove** button to remove the ticked projects from the user's list.

To add a new *project*, select it from the drop down. Note that the way the drop-down is formatted for adding new *projects* will change depending on the number of *projects* that exist. For a small number of *projects*, you get a simple drop down. When the number grows, then you get both a drop-down menu with all *projects*, but also a text box in which to type the *project* name. As the name is typed, it shows a selection of *projects* that match.

The *user's association of projects* is only changed once the **Save** button is pressed, in the regular button bar. Note that the *projects* themselves are also changed because they know which users are associated with them.

Changing A User's User Groups

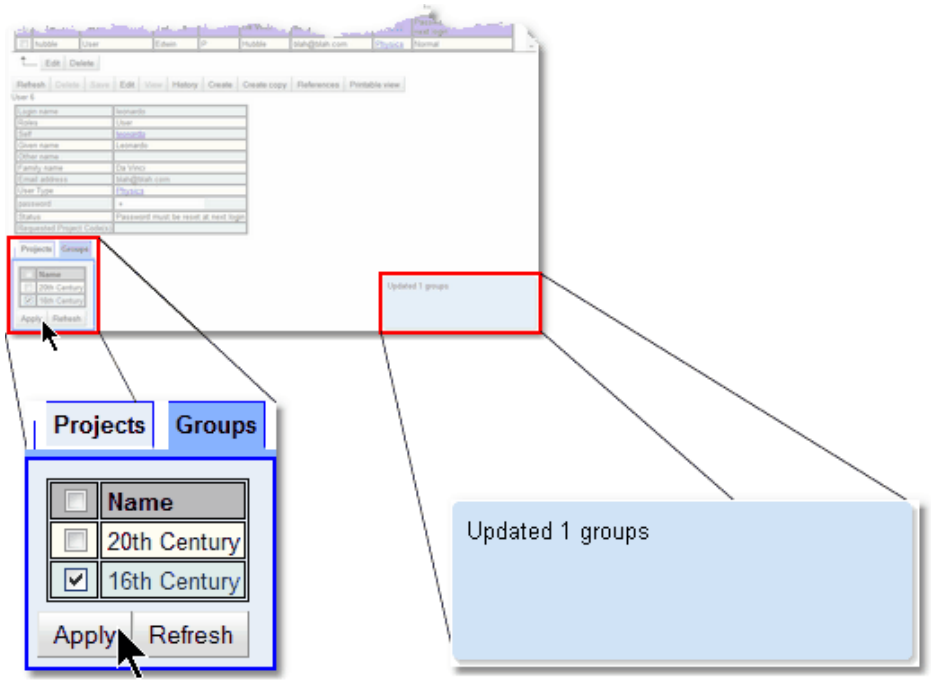
To see which *user groups*[450] a user is a member of select the user from the list of users, as described by How To Edit A Single User[130] above. The tab panel at the bottom of the page shows the user's *projects* and *group* memberships. Select the **Groups** tab and a list of all *user groups* is seen, with a check box next to each group the user is a member of.



© 2010-2015 Exprodo Software

Note: the user's details do not need to be in edit mode in order to change the *user group* membership. That's because the *user group* membership is not a *property* of the user itself, but of the *user groups*. To modify the *group* membership for the selected user:

1. Tick or untick the groups. The *groups* ticked are the ones the user is a member of. Tick the check box in the header row if all the check boxes are to be ticked or unticked.
2. Press the **Apply** button when finished.



Configuring The Properties Displayed

The user can control which *properties* [448] are displayed in the list of items, in the detailed view of a item. This is done in the Bakery [351].

- Properties Visible In A Biskit List [361] explains how to change the *properties* that are displayed in a list of items. Change the visibility of each of the *properties* defined on an item.
- Properties Visible In Biskit Detail [361] explains how to change the *properties* that are displayed when seeing a detailed view of an item. In this case, it affects both the read-only and editable mode detailed view of an item. Again, change the visibility of each of the *properties* defined on an item.
- Properties Visible In A Collection Editor [361] explains how to change the *properties* that are displayed in collection editors. For example, if there is a *Project* Biskit Type [446] that contains a list of users, then the user can change which *properties* are shown to represent those users in the *Project's* list of users.

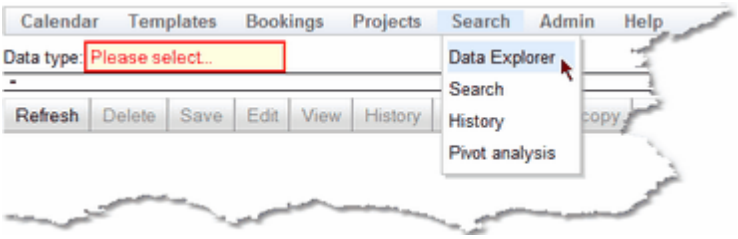
© 2010-2015 Exprodo Software

4.3.7 Changing A User's Settings

A user can change their own *settings*[450], as described in User Settings[32] in the **Calpendo User Guide**. Sometimes, however, the administrator may want to edit another *user's settings*. This can be desirable if some users do not have *permission*[448] to modify all of their *settings*. For example, **Calpendo** may have been configured so that a regular user cannot choose which menu they use, whereas an administrator may have *permission* to do so. The menu is one of the *properties*[448] of a *user's settings*.

Page For Editing Another User's Settings

Currently, there is no page dedicated to editing other *user's settings*. To do this, use Data Explorer[127] and set the *Biskit Type*[446] to **Calpendo** *User Settings*. By default, Data Explorer[127] appears on the menu here:



4.3.8 Special User

There is only one special user, and that is **nobody**. This is a user whose *status*[450] should be set to **Blocked** and never used. This user is used in the following circumstance:

Sending automated emails with embedded data:

When an Automatic Email[297] is sent, it can include data taken from some *Biskits*[446]. When this happens, to make sure that it is safe to include such information, make sure that the information is only readable by whomever the email goes to, or whomever initiated the action that caused the email to be sent, as it is possible that anybody could read the email. It is therefore better to ensure that emails only include information that is readable by the special user **nobody**. If the special user **nobody** can read the data, then it's safe to put it into an email. Your administrator should have set up which data **nobody** has Permission[278] to read.

Please note that, while there are users created by default that are called **Root** and **Admin**, these are **not** special users. They have a *user role*[450] of **Root** and **Admin** respectively, and their *roles* are special, but the users are not. As many users with the **Root** role may be created as required. Any user with the **Root** role is not subjected to *permissions*[448] checks, regardless of whether it is the user **Root** or not; only the *roles* matter.

© 2010-2015 Exprodo Software

4.4 Bookmark Manager

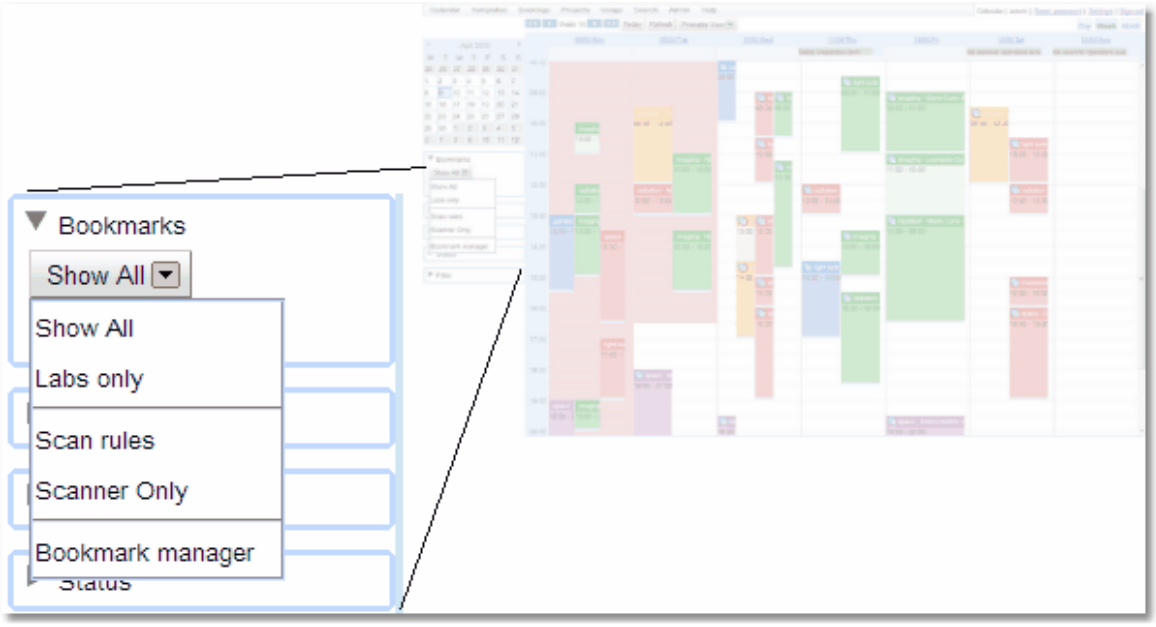
The **Bookmark Manager** allows *bookmarks*[447] to be created/edited for use in the *Bookings Calendar*[446]. The manager also keeps track of which *bookmarks* apply to which users and which *bookmarks* are system wide.

A *bookmark* holds a list of *resources*[449] that a user would like to see on the *calendar* at the same time. This is a quick way for the user to set up their *calendar* to view the required *resources*. A user may have many *bookmarks* to enable different views of the *calendar*. A user may see their own *bookmarks* and all system wide *bookmarks*. When a user creates a *bookmark* they can make them system wide for other users to use.

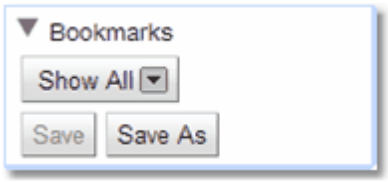
All users start with the default **System Bookmarks->Show All** *bookmark* by default. It is important that if you create any new resources[270] you add them to that *bookmark* so all users can see them initially.

Calendar Page Bookmarks

This is what a standard *bookmark* list would look like in the *calender* view. The top two *bookmarks* are system wide, the bottom two are the users specific *bookmarks*, the final option opens up the **Bookmark Manager** for the user:

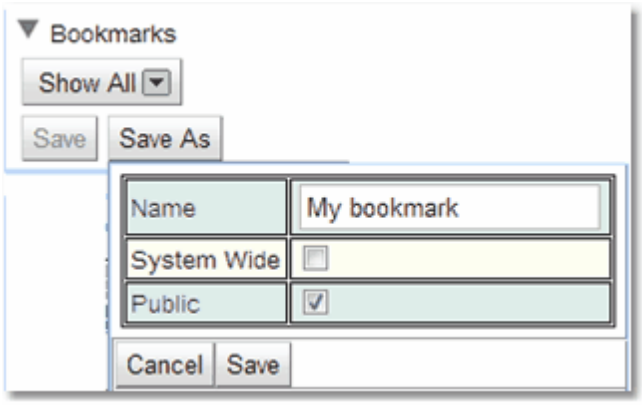


A user with the correct *permission*[448] may save a current *resource* configuration as a bookmark. Set up the *resources* required in the *bookmark* as those that can be seen on the *calendar*, under the *bookmark* pull down, click on the **Save As** button.



© 2010-2015 Exprodo Software

Then save the *bookmark*, giving it a name, saying whether anyone can see it (**System Wide**). The **Public** *property* 448 is not currently used (any user can edit this *bookmark*).



User Bookmark Permissions

By default non-admin users cannot create *bookmarks* or access the **Bookmark Manager**, either from the **Admin->Bookmark Manager** menu option or the **Bookmark Manager** option on the bottom of their *bookmark* list in the *calendar* view. In order to change this *Permission* **Create->Bookmark->Anybody can create non-system bookmarks** would need to be enabled. If you have given users *permission* to create *bookmarks* you will need to give them *permission* to edit *bookmarks* by enabling **Update->Bookmark->Anybody can update their own non-system bookmarks**. Please read the Permissions 60 chapter for more information on setting and using *Permissions*.

Bookmark Manager

To open up the **Bookmark Manager** the **Admin** user can use the **Admin->Bookmark Manager** item. A user would have to use the option on the pop up from the *Bookmarks* section of the *Calendar* page (see **Calender Page Bookmarks**).

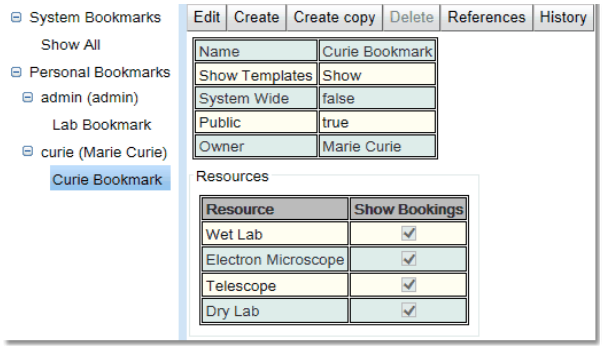
The **Bookmark Manager** allows users to see their own and system wide *bookmarks*. If they have the permissions the user may edit and delete these *bookmarks*. An **Admin** user can see all users personal *bookmarks* as well as all the system wide *bookmarks*, and may edit or delete any of them. The exception is no-one can delete a *bookmark* that is currently in use.

© 2010-2015 Exprodo Software

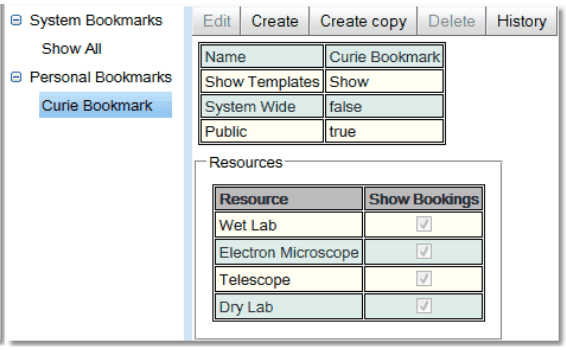
A *bookmark* has a number of basic *properties*[448]:

Property	Description
Name	The name of the *bookmark* as assigned by the user.
Show Templates	If you are in the *calendar* view which *Time Templates*[449] will be shown. There are two choices: **Show**: Show any *Time Template* information in the *Bookings Calendar*. **Hide**: Hide any *Time Template* information in the *Bookings Calendar*.
System Wide	Can everyone see the report or is it a personal one. If this is true the report will appear under **System Bookmarks** rather than **Personal Bookmarks**.
Public	If a system wide *bookmark* can anyone modify it. (This is currently not being used).
Owner	The user who created the *bookmark* and if **Public** is false, the only person who can edit a *bookmark* if it is also **System Wide.**

The page is split into two. On the left is a list of the **System Bookmarks** (those that anyone can see) and the users **Personal Bookmarks**. On the right is the area where *bookmarks* can edited, created and deleted. Where the page splits can be moved to change the space available for each of the sides.



Admin view of the Bookmark Manager



User View of the Bookmark Manager

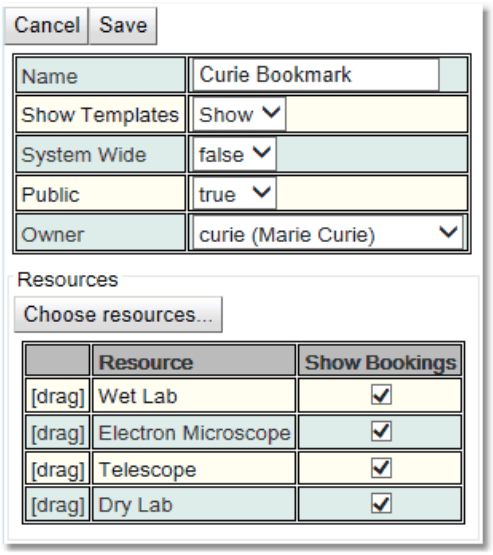
The main difference between the two views is that the **Admin** user can see **Personal Bookmarks** for all users. Also the **Admin** user can see the *owner*[448] of a *bookmark* and also get a list of *references*[449] to the *bookmark*.

For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition[96] chapter.

© 2010-2015 Exprodo Software

Editing Bookmarks

Once the choice to edit a *bookmark* or create a new one has been made the view will be in edit mode.



In the top half of the view there are the *properties* that can be edited. When finished either **Save** to complete or **Cancel** to reject the changes.

In the bottom half of the view the *resources* that will be shown by using this *bookmark* can be chosen. To add or delete resources available in this list use the **Choose Resources...** button. This will open up a resource selector. To learn more about how this works read the section on the Resource Selector. Once the *resources* required are in this list, check/uncheck the *resources* required to view in the *Bookings Calendar* for this *bookmark* .

Click on the first column *[drag]* to drag and drop *resources* to reorder the list.

© 2010-2015 Exprodo Software

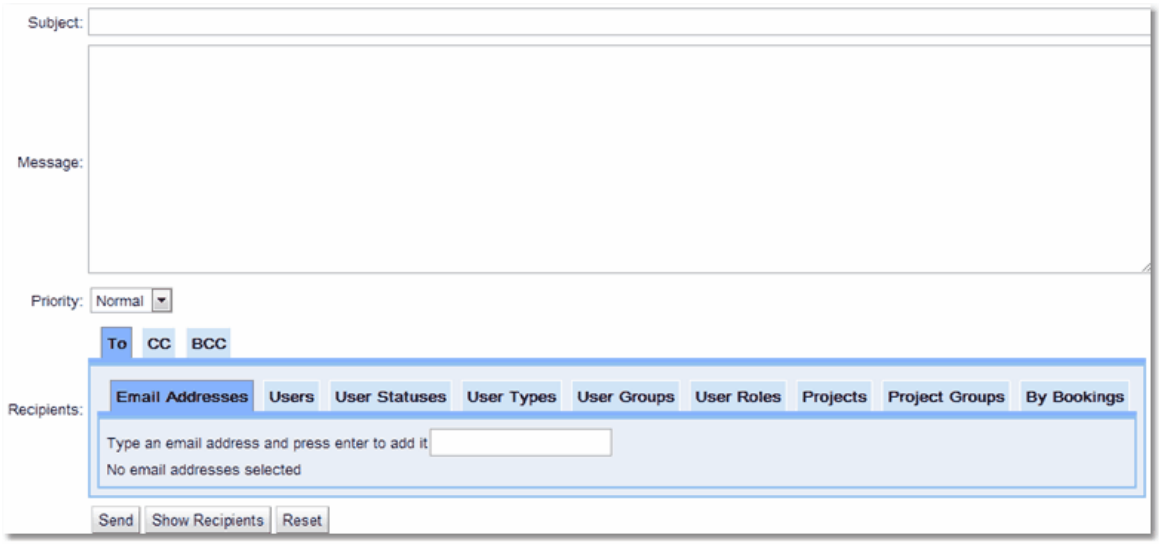
4.5 Using Calpendo To Send Emails

Calpendo can be used to send emails to anybody that is registered with **Calpendo**, since all users must provide an email address. Emails can also be sent to any email address as required. This can be used as a means of general communication so that the administrator doesn't necessarily need to maintain email distribution lists. This works because if you want to send emails to people who use your *resources*[449], then they are probably also **Calpendo** users. If you are running **Calpendo** on your own server, **Calpendo** can make a secure connection to the mail server for sending email see Global Preferences Email on how to set this up..

Calpendo can also be used to send email to people who have a *booking*[446] over the next few days, or whatever period required. This is useful when there's a problem with a *resource* that needs to be brought to the attention of those that are due to use it.

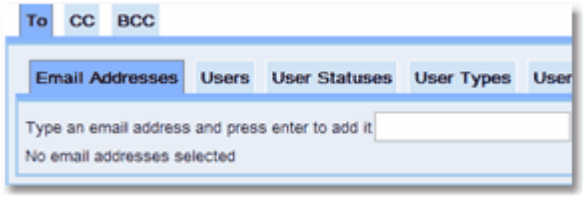
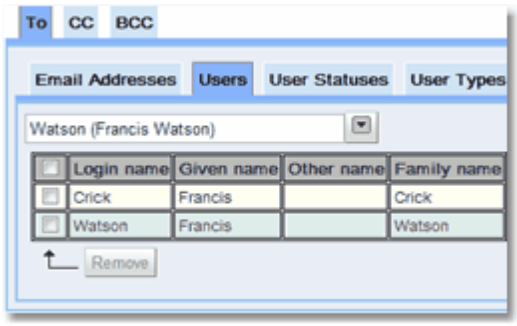
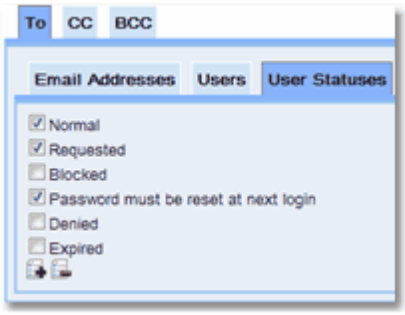
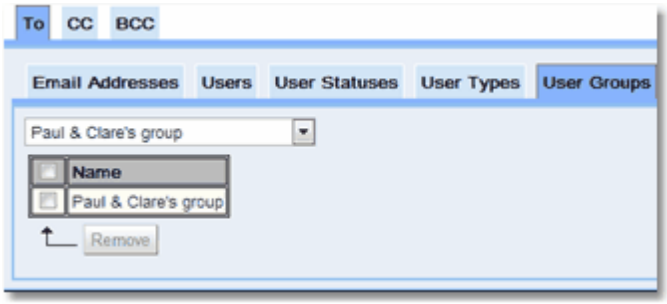
Send emails using the **Manual Email** page, which is placed by default onto the menu as **Admin-->EMail-->Manual Email**.

The page sets up the email subject and email message and priority, as though using any normal email client. There are then three tabs, named **To**, **CC** and **BCC**, to specify who the email is sent to, copied to and blindly sent to. As long as someone is in the **CC** or **BCC** list then there doesn't need to be anything in the **To** list.

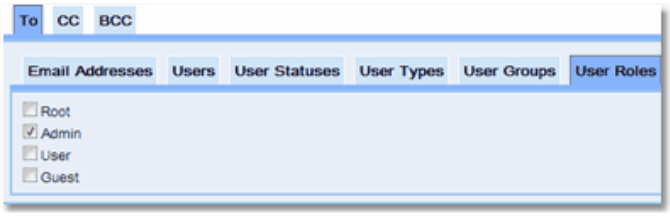
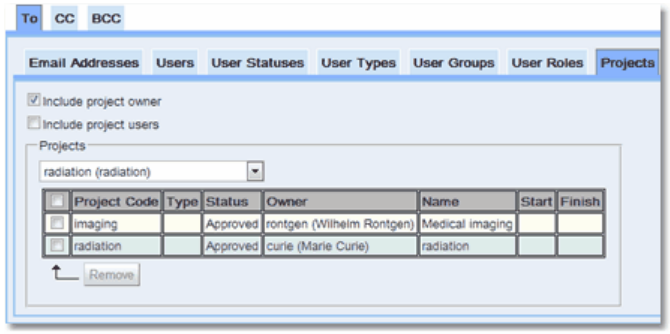
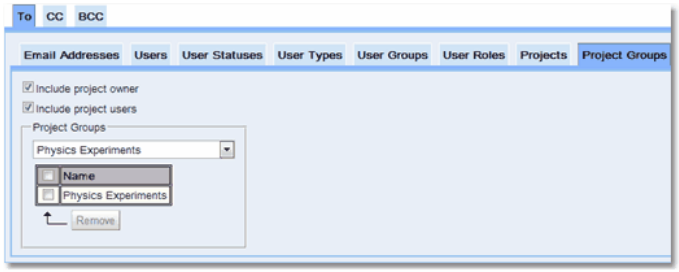
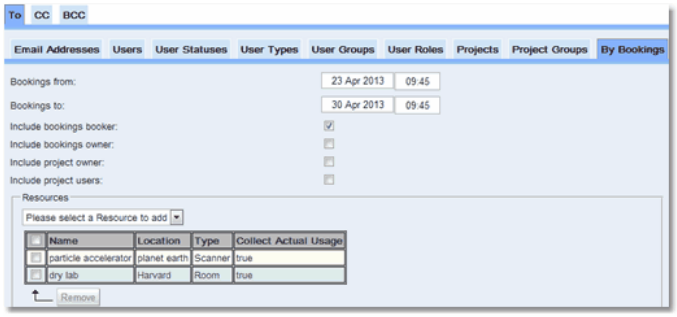


© 2010-2015 Exprodo Software

Each of the **To**, **CC** and **BCC** tabs have the same content, with tabs as follows:

Tab Name	Description	
Email Addresses	This adds any number of specific email addresses. This means emails can be sent to people who are not known by **Calpendo**.	
Users	This chooses individual users to send the email to. Use the drop down to select the users required. Use the tick box to choose which users you wish to remove and then click the **Remove** button.	
User Statuses	This chooses the *statuses*[450] of users that are allowed to received the email. Note that the recipients will be those specified by any of the other tabs that are not also prohibited by this **User Statuses** tab.	
User Types	If **Calpendo** is configured so that users are classified by their *user type*[450], then select one or more *user types* so that all users with those *users types* should receive an email.	
User Groups	Select any number of *user groups*[450] to send email to. Use the drop down to select the *user groups* required. Use the tick box to choose which *user groups* to remove and then click the	

© 2010-2015 Exprodo Software

Tab Name	Description	
	Remove button.	
User Roles	Add any user to the recipient list that has the selected *roles*[450].	
Projects	Select any number of *projects*[448], and choose whether the email should be sent to the *projects'* owners, their users or both. Use the drop down to select the *projects* required. Use the tick box to choose which projects to remove and then click the **Remove** button.	
Project Groups	Similar to the **Projects** tab, this chooses *project groups*[448] and whether the email should be sent to the *projects'* owners, their users or both.	
By Bookings	This specifies a selection of *bookings* by the *resource* and time period, and then to choose whether the email should go to the person that made each *booking*, the person that owns it, and for any *project* associated with the *booking* to choose whether to send the email to the *project* owner or the *project* users.	

© 2010-2015 Exprodo Software

When the email and its recipients are ready, then use the buttons in the button bar at the bottom:

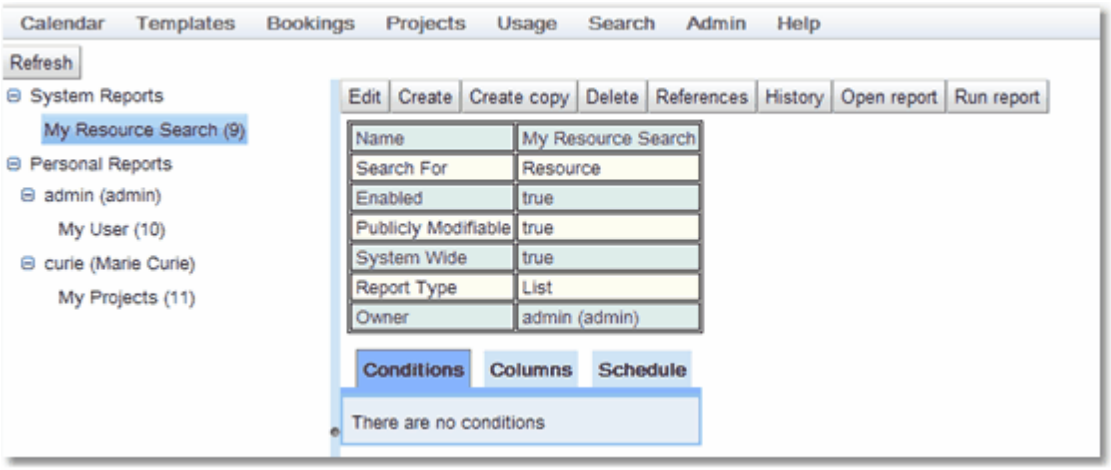
Button	Description
Send	This sends the email
Show Recipients	**Calpendo** calculates who the email would be sent to and shows a list of those users without actually sending the email. Use this to verify that the email will target the expected users. If an email address appears more than once in any of the tabs then only one email is sent with To overriding CC overriding BCC to decide which address line is used.
Reset	After using the **Manual Email** page, press the **Reset** button to reset the page back to its original content, ready to compose a new email.

© 2010-2015 Exprodo Software

4.6 Report Manager

The **Report Manager** allows an **Admin** user to see reports for all users as well as the system reports.To access the **Report Manager** page it is found on the **Admin** menu. However, the administrator may have configured **Calpendo** so that the menu is different.

The page is split into two. On the left is a list of the **System Reports** (those that anyone can see) and all the **Personal Reports**. On the right is the area where **Reports** can edited, created and deleted. Where the page splits can be moved to change the space available for each of the sides.



In order to understand how to create and edit reports read the the Reports[121] section of the **Calpendo User Guide**.

© 2010-2015 Exprodo Software

4.7 Import

Calpendo supports the import of data using comma-separated values (csv). This is found on the **Admin->Import** menu.

In order to do this an import file needs to be created. Only information for a single *Biskit Type* [446] can be imported at a time. For example if importing *projects* [448], import the *project* and then separately the *project resource settings* [448] for each of the *projects*. The exception to this is, if importing data belonging to a **Slave** *Biskit Type*, that needs to be imported with the **Master** *Biskit Type* data as if the **Slave** *properties* [448] actually belonged to the **Master** rather than the **Slave**.

Once the import files have been created they can be used in the **Calpendo** import page to import your information.

4.7.1 Example Import Files

The file should have a header row which defines the *properties* [448] that are going to be imported, and then one row for each object of the *Biskit Type* [446]. Use the Search [108] page or the Bakery [351] to find the names for the *properties*. All the fields need to be comma separated.

To modify the value of a *property* that is from a component *biskit* [446], values to all the *properties* on that component must be provided. Any *properties* not mentioned will be attempted to be set to null. If this fails then an error will occur.
Eg, When importing users and setting a new value for **userIdentity.authenticationMethod**, then a value for **userIdentity.loginName** must also be set.
If a new value for **userIdentity.authenticationMethod** is set and a new value for **userIdentity.loginName** is not set, then the **userIdentity.loginName** will be set to null.

A *property* that is used to identify the *biskit* to be updated can itself be updated. For example, when importing users, users could be identified by their **email** *property* (if it's unique) and the import could also modify that **email** *property*. To do this, two **email** columns in the CSV file are required.

i.e.
email,email
tim@conaptic.com,timb@conaptic.com

The first column would then be the key *property* to identify the user to be changed and the second column would be the new value of the **email** *property*.

Use the following options **Key:** and **Ignore:** (typed before the property name) to pre select the properties that will be key fields or fields to be ignored.

© 2010-2015 Exprodo Software

Example File For Importing A User

The **User** record that will be created from this file:

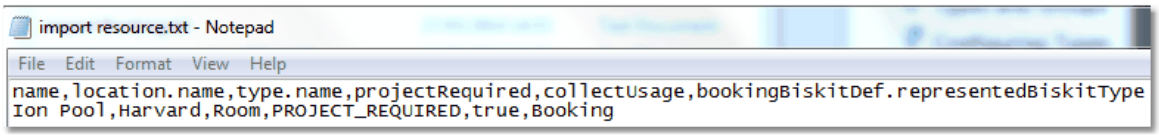
Login name	Timb
Roles	User
Self	Timb (Tim Bilderbeck)
Given name	Tim
Other name	
Family name	Bilderbeck
Email address	blah@conaptic.com
User Type	Physics
Password	••••••
Status	Normal
Expiry Date	
Requested Project Code(s)	

Name	Value	Comments
userIdentity.loginName	Timb	String
userIdentity.authenticationMethod	Local	Needs to match available authentication method exactly.
roles	4	Set by bits 1 = **Root** 2 = **Admin** 4 = **User** 8 = **Guest** See User Roles 368 for more information.
givenName	Tim	String
familyName	Bilderbeck	String
email	blah@conaptic.com	String
userType.name	Physics	Needs to match name exactly
password	timbil	Depends on system configuration
status	Normal	Requested, Normal, Blocked

Notice the order of the *properties* does not matter. Depending on how the system is configured password may be a compulsory *property* and may have a minimum number of characters and/or a combination of alphanumerics.

© 2010-2015 Exprodo Software

Example File For Importing A Resource



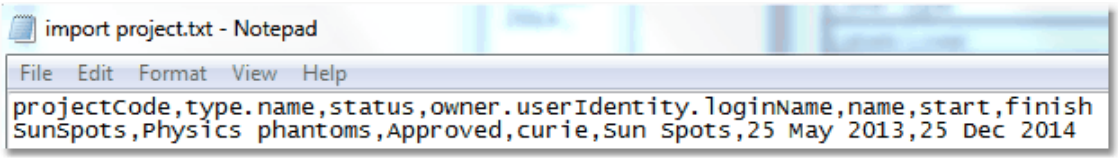
The **Resource** record that will be created from this file:

Name	Ion Pool
Location	Harvard
Type	Room
Project Required	Any Project Required
Booking Subtype	Booking
Require Reason for Cancellations	☐
Allocate Calendar Column	☑
Collect Actual Usage	☑
Allow Old Changes	☐
Enable pre-defined time slots	☐
Version	3
Created	19 May 2014 09:26
Updated	21 May 2014 10:12

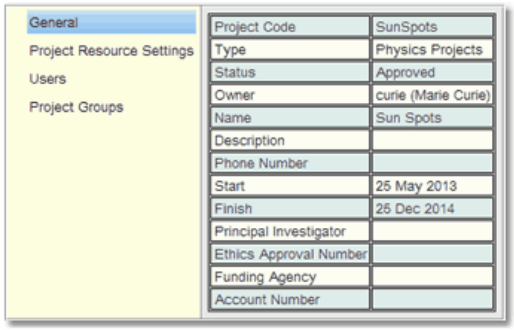
Name	Value	Comments
name	Ion Pool	String
location.name	Harvard	Needs to match name exactly
type.name	Room	Needs to match name exactly
projectRequired	PROJECT_REQUIRED	PROJECT_REQUIRED or PROJECT_NOT_REQUIRED or PROJECT_WITH_RESOURCE_ENTRY
collectUsage	true	true or false
bookingBiskitDef.representedBiskitType	Booking	The Biskit Type to be used for bookings of this resource

© 2010-2015 Exprodo Software

Example File For Importing A Project


import project.txt - Notepad
File Edit Format View Help
projectCode,type.name,status,owner.userIdentity.loginName,name,start,finish
SunSpots,Physics phantoms,Approved,curie,Sun Spots,25 May 2013,25 Dec 2014

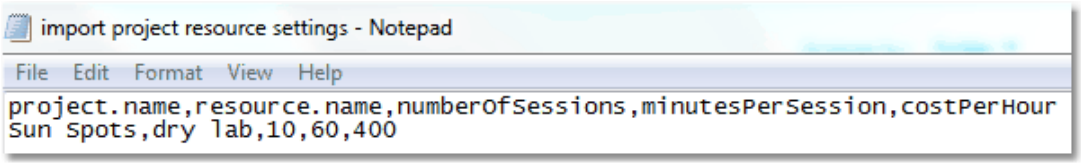
The **Project** record that will be created from this file:

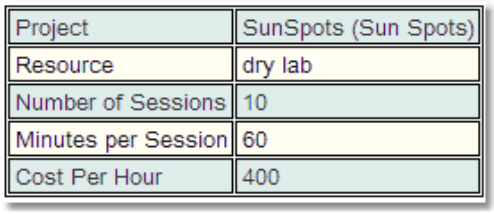


Name	Value	Comments
projectCode	SunSpots	String
type.name	Physics Project	Needs to match name exactly
status	Approved	Approved, Requested, Denied, Terminated
owner.userIdentity.loginName	curie	Login name
name	Sun Spots	String
start	25 May 2013	Date
finish	25 Dec 2014	Date

Projects have other information attached to them but they are stored in different *Biskit Types*. These are used to store the *Project Resource Settings* [448], **Users** and *Project Groups* [448] *properties*. In order to import this information, first import the *project* information and then import each of the other three *Biskit Types* one after another.

For example importing the *Project Resource Settings* for the above project:

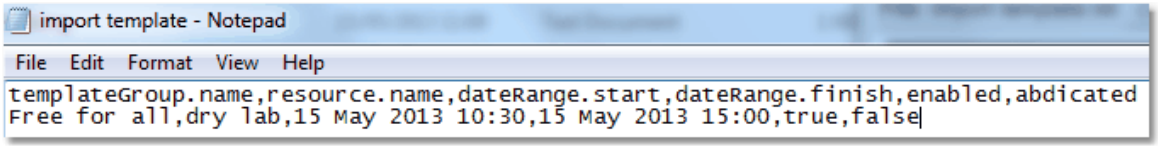




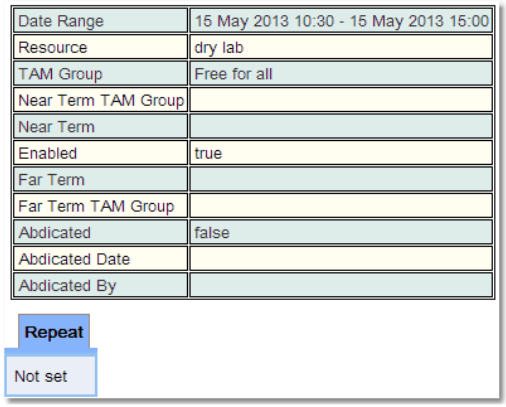
Name	Value	Comments
project.name	SunSpots	Needs to match name exactly
resource.name	Physics Project	Needs to match name exactly
numberOfSessions	10	Integer
minutesPerSession	60	Integer
costPerHour	400	Double

© 2010-2015 Exprodo Software

Example File For Importing A Time Template


import template - Notepad
File Edit Format View Help
templateGroup.name,resource.name,dateRange.start,dateRange.finish,enabled,abdicated
Free for all,dry lab,15 May 2013 10:30,15 May 2013 15:00,true,false

The **Template** record that will be created from this file:

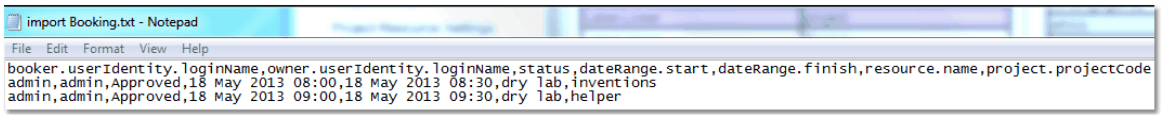


Name	Value	Comments
date Range.start	15 May 2013 10:30	Date and Time
dateRange.finish	15 May 2013 15:00	Date and Time
templateGroup	Free for All	string
Enabled	true	true or false
Abdicated	false	true or false

To import repeat *Time Templates* see below for additional information about importing *repeat bookings*.

© 2010-2015 Exprodo Software

Example File For Importing Two Bookings


import Booking.txt - Notepad
File Edit Format View Help
booker.userIdentity.loginName,owner.userIdentity.loginName,status,dateRange.start,dateRange.finish,resource.name,project.projectCode
admin,admin,Approved,18 May 2013 08:00,18 May 2013 08:30,dry lab,inventions
admin,admin,Approved,18 May 2013 09:00,18 May 2013 09:30,dry lab,helper

The **Booking** record that will be created from the first import line in this file is:

Resource	dry lab
Type	
Booker	admin (admin)
Project Resource Settings	dry lab 0 scans of 60 mins for £300.0
Owner	admin (admin)
Project	inventions (inventions)
Status	Approved
Approved by template	false
Warned	
Date Range	18 May 2013 08:00 - 08:30
Created	16 May 2013 10:23
Cancelled	
Modified	
All day	
Description	
Reminder Trigger Time	
Reminder Sent	false
Remind Booker	false
Remind Owner	false
Remind Project Owner	false
Remind Project Users	false

Repeat

Not set

Name	Value	Comments
booker.userIdentity.loginName	admin	Needs to match login name exactly
owner.userIdentity.loginName	admin	Needs to match login name exactly
status	Approved	**Approved, Requested, Cancelled, Denied**
dateRange.start	18 May 2013 08:00	Date and Time
dateRange.finish	18 May 2013 08:30	Date and Time
resource.name	dry lab	Needs to match name exactly
project.ProjectCode	inventions	Needs to match project code exactly

© 2010-2015 Exprodo Software

When importing *repeat*[449] *bookings* the repeat sub *Biskit Type* for the *repeat booking* needs to be defined as well as all the *properties* defined for that *Biskit Type* and all the properties of the parent *Biskit Type* **Repeat**. Any *property* not defined will have its value set to null.

Biskit Type	Property	Type	Example Value
Repeat	repeat.biskitType (must always be present)	Biskit	AnnualRepeat DailyRepeat MonthlyRepeatByDate MonthlyRepeatByDay WeeklyRepeat
	repeat.start	Date	12 January 2015 13:00
	repeat.finish	Date	12 January 2999 14:00 2999 for never ending repeats
	repeat.repeatEvery	int	1
	repeat.repeatType (must match repeat.biskitType)	JavaEnum	Annually Daily Monthly by date Monthly by day Weekly
Annual Repeat	repeat.dayOfMonth	Int	1-31
	repeat.monthOfYear	int	0-11 with 0 = January
Daily Repeat	None		
Monthly Repeat By Date	repeat.dayOfMonth	int	1-31
Monthly Repeat By Day	repeat.dayOfWeek	int	1-7 with 1 = Sunday
	repeat.weekOfMonth	int	1-5
Weekly Repeat	repeat.sunday	Boolean	True or False
	repeat.monday	Boolean	True or False
	repeat.tuesday	Boolean	True or False
	repeat.wednesday	Boolean	True or False
	repeat.thursday	Boolean	True or False
	repeat.friday	Boolean	True or False
	repeat.saturday	Boolean	True or False

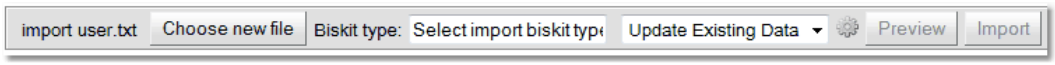
© 2010-2015 Exprodo Software

4.7.2 The Import Page

When importing data you can specify whether data is imported if there are any errors or that no data is imported from the file if any errors occur, errors include *Time Templates*[449], *Booking Rules*[446] and *Permissions*[448] failing to allow creation or updating of records. Examples of how different error types are handled can be found in the section on Handling Import File Errors[187].

On entering the **Import** page it will look like:

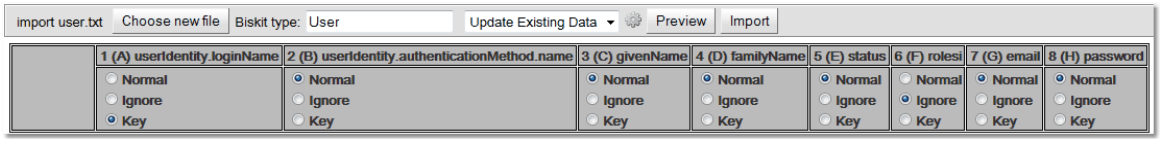
Press **Browse...** (or **Choose File**) to select the file to import. At this point **Calpendo** will attempt to decide which *Biskit Type*[446] is being imported from the header information. If it cannot determine the *Biskit Type*, the user will need to select it.



Once the **Import** page knows which *Biskit Type* it is importing it will look like this:

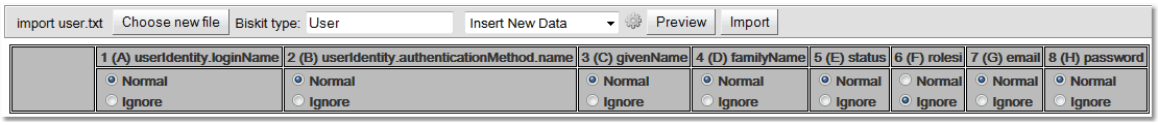
At this point decide whether to choose to **Update Existing Data** or **Insert New data**. If updating data then import needs to know which columns to use as the key fields to find the appropriate records to update. Click on the **Key** option below the column headings of those *properties*[448] that are keys fields or put **Key**: before the *property* name in the file.

In this example for importing **User** data, the key is specified to be the combination of **userIdentity.loginName and userIdentity.authenticationMethod.name**, as the combination of these two will be unique.



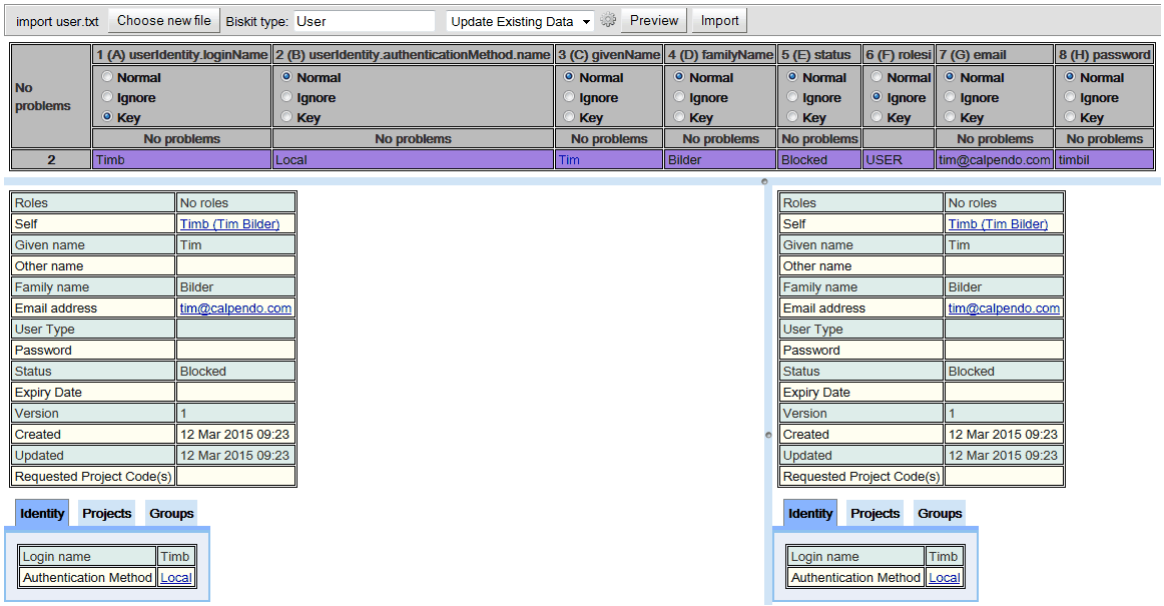
© 2010-2015 Exprodo Software

In this example for importing **User** data, **Insert New Data**. has been chosen, the **Key** option has disappeared as no longer required.

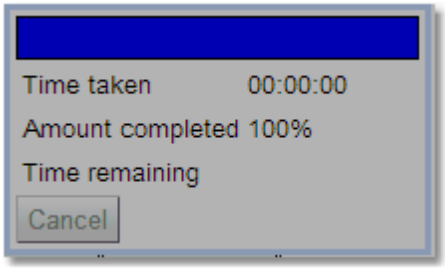


Choose the columns to be ignored during the import by clicking the **Ignore** option below the *property* name or by putting **Ignore:** before the *property* name in the file, and then press the **Preview** button to see the data before importing or go straight to importing by pressing the **Import** button.

If previewing once the data has been loaded into the top pane, select an individual record and see the record to be created/updated in the bottom left pane, and if doing an update the original record is in the bottom right pane.



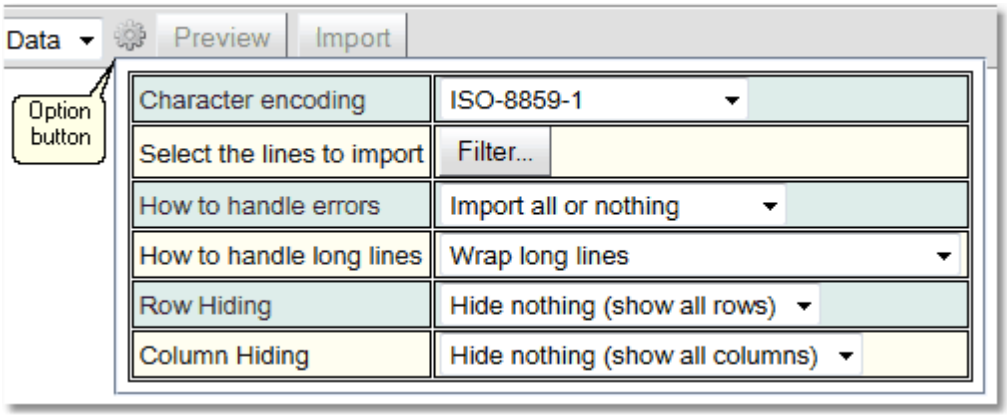
Press the **Import** button to import the data into **Calpendo**. While the import is being run a progress bar will be displayed which will show the time taken, % amount completed and any errors. There is also a **Cancel** button to cancel the import. When the import completes a pop up will appear in the bottom right hand corner of the screen saying **Import Completed**.



© 2010-2015 Exprodo Software

Importing Options

Click on the options button to see all the options available for viewing and importing the data file

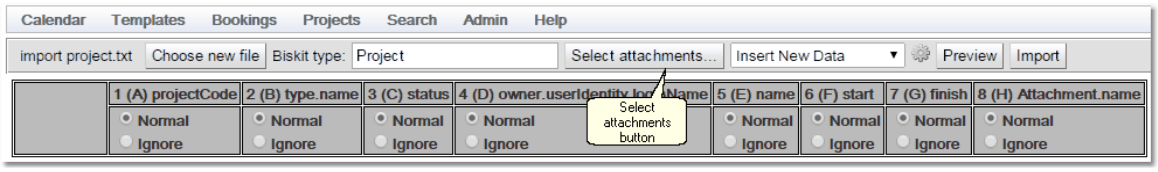


Option	Description
Character encoding	While importing Calpendo guesses at the character encoding used for the file being imported, and allows the administrator the chance to choose the one to use, allowing non-ASCII files to be imported.
Select the lines to import	Use the **Filter...** button to set up *conditions* 447 to define which rows will be imported.
How to handle errors	Choose **Import any that do not fail** which will import all records that do not have errors, or **Import all or nothing**, which will only import the records if they would all succeed.
How to handle long lines	Whether lines should be wrapped making the box deeper, **Wrap long lines,** or whether to make the table wider to deal with long lines. **Do not wrap long lines; make the table wider.**
Row Hiding	Whether to hide those rows that are being ignored for import, **Hide ignored rows**, or to hide all rows without an error, **Hide error-free rows**. Press the **Preview** button to see the results of any changes to this setting.
Column Hiding	Whether to hide those columns that are being ignored for import, **Hide ignored columns**, or to hide all columns without an error, **Hide error-free columns**. Press the **Preview** button to see the results of any changes to this setting.

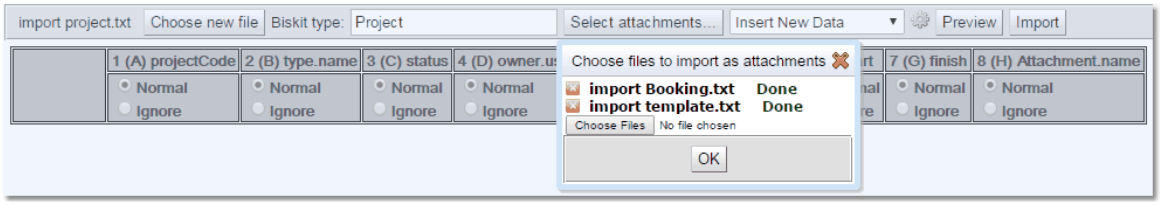
© 2010-2015 Exprodo Software

Importing File Attachments

If any of the *properties* to be imported include the **Attachment** *BiskitDef*[446], then the user is supplied with an additional button to select the list of attachments to be imported. This needs to be done before the **Preview** or **Import** buttons are pressed.



Once the attachment pop up is present, use the **Browse...** (or **Choose File**) button to find all the attachments required for the import. Once selected click OK and move on to **Preview** and **Import** the data.



Handling Import File Errors

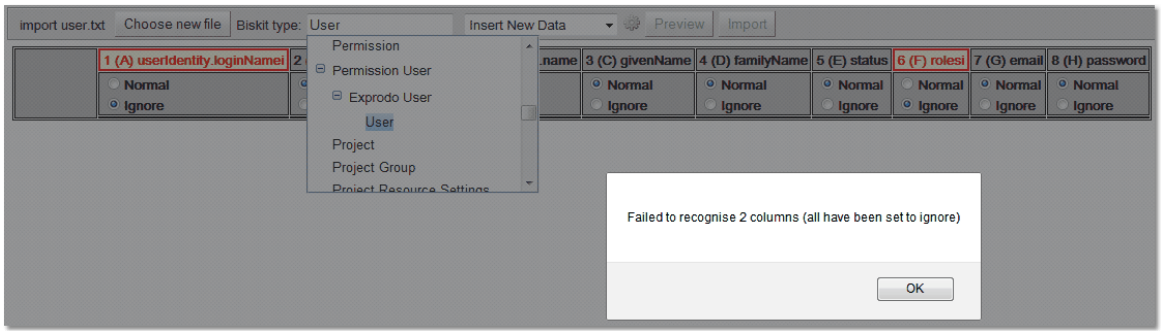
During the import process there are three main types of errors.

1. Incorrect *property* names or labels in the header of the import file. (Header errors)
2. Incorrect data in a record, stopping **Import** from resolving property values. (Data Errors)
3. Failure to import due to internal checking. (Violation errors)
    a. Incorrect data in a record, causing *Booking Rules* or *Permissions* to fail to allow creation of records(e.g. password too short).
    b. *Time Templates*, *Booking Rules* and *Permissions* failing to allow creation or updating of records.

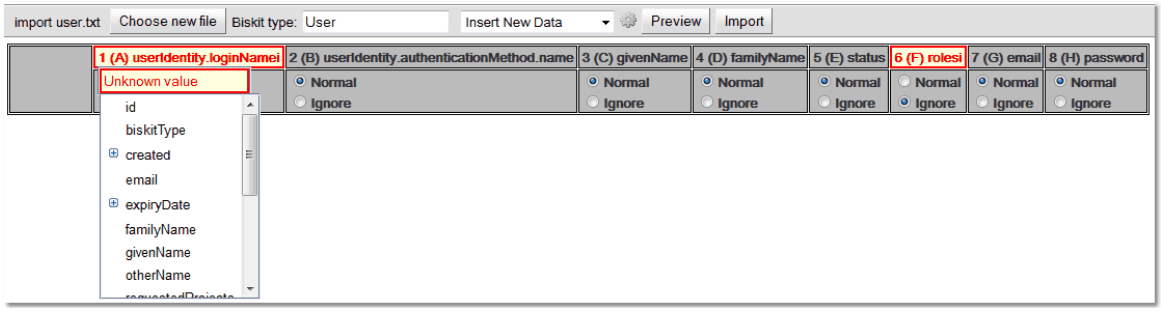
© 2010-2015 Exprodo Software

Header Errors

Header errors are found when the import file is first loaded using the **Send** button. Once **Import** has decided on the *Biskit Type* being imported or the user has selected the correct *Biskit Type*, the header information is loaded. If any of the headers cannot be resolved to *property* names or *property* labels (in a case insensitive manner if possible) then those headers are marked in red. A pop up will inform the user how many column headings have not been recognised. Import then updates those columns from **Normal** to **Ignore**, so the user can proceed with the import without those columns.



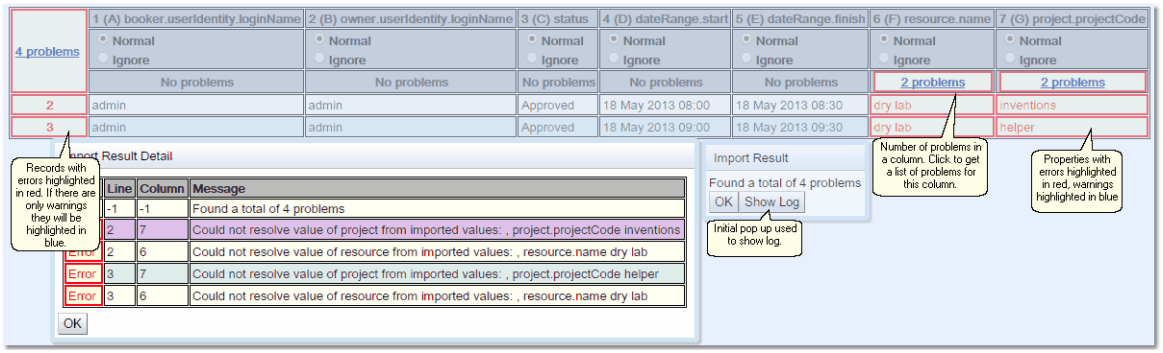
These errors can be fixed in the Import page. Just click on any red header and you will get a drop down box with a complete lists of this *Biskit Types properties*. Select the correct one, and the error will be removed. Once done the column will be moved from **Ignore** to automatically.



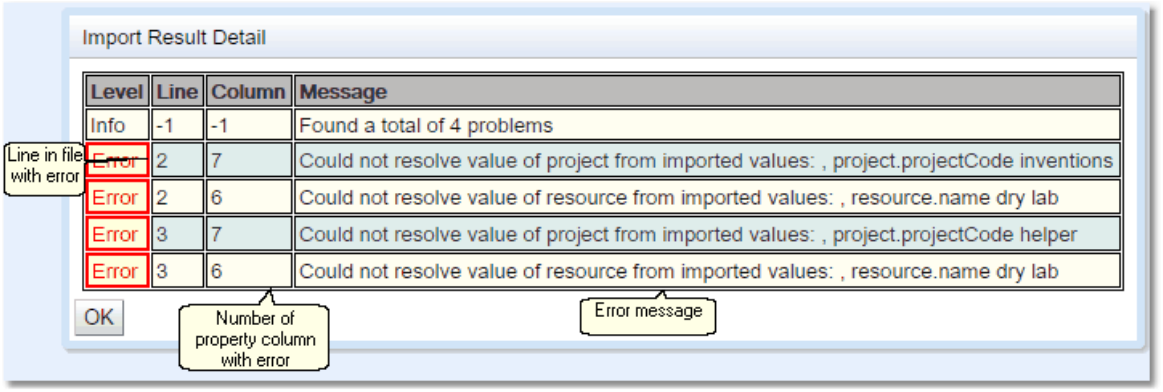
© 2010-2015 Exprodo Software

Data Errors

Data errors are found either during the preview of the data, of if a preview is not done, during import.


4 problems
1 (A) booker.userIdentity.loginName
2 (B) owner.userIdentity.loginName
3 (C) status
4 (D) dateRange.start
5 (E) dateRange.finish
6 (F) resource.name
7 (G) project.projectCode
Normal
Ignore
No problems
2 problems
admin
Approved
18 May 2013 08:00
18 May 2013 08:30
18 May 2013 09:00
18 May 2013 09:30
dry lab
inventions
helper
Import Result Detail
Line
Column
Message
Found a total of 4 problems
Could not resolve value of project from imported values: , project.projectCode inventions
Could not resolve value of resource from imported values: , resource.name dry lab
Could not resolve value of project from imported values: , project.projectCode helper
Error
OK
Import Result
Found a total of 4 problems
Show Log
Records with errors highlighted in red. If there are only warnings they will be highlighted in blue
Initial pop up used to show log.
Number of problems in a column. Click to get a list of problems for this column.
Properties with errors highlighted in red, warnings highlighted in blue

If there are problems a pop up will appear use the **Show Log** button to show a complete list of the problems. Records with an error will have their record number highlighted in red as will those *property* values which are causing an error. If a record has only warnings it will highlighted in blue, as will those *property* values which are just warnings. Moving the cursor over a *property* with an error/warning will result in a tool tip describing the error/warning. Also for each column you can see the number of problems. If there are any problems in a column clicking on the number of problem text will produce a pop up which will show the problems for just that column.


Import Result Detail
Level
Line
Column
Message
Info
-1
Found a total of 4 problems
Error
2
7
Could not resolve value of project from imported values: , project.projectCode inventions
6
Could not resolve value of resource from imported values: , resource.name dry lab
3
Could not resolve value of project from imported values: , project.projectCode helper
Could not resolve value of resource from imported values: , resource.name dry lab
OK
Line in file with error
Number of property column with error
Error message

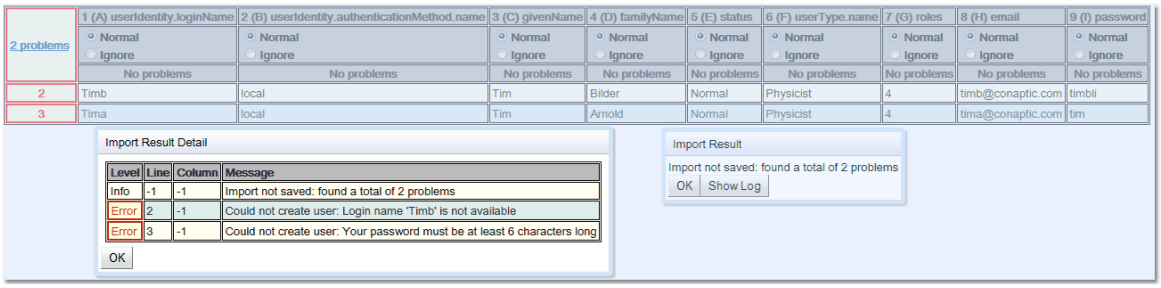
The problem log will list theproblems, showing the line and column the problem occurs on as well as the problem message. Clicking on the an errors column will scroll the file to the error, and it will be coloured blue to stand out.

In order to fix these types of problems go back to the original import document and fix them there before loading it back into the **Import** page. Remember once the file to be imported has been updated, save it before reloading the file into **Calpendo Import**.

© 2010-2015 Exprodo Software

Violation Errors

Violation errors are found during the import of the data into the database. These are errors that occur when the **Calpendo** is trying to create or update the database. During this process all the *Time Templates, Booking Rules* and *Permissions* are checked.


2 problems
1 (A) userIdentity.loginName
2 (B) userIdentity.authenticationMethod.name
3 (C) givenName
4 (D) familyName
5 (E) status
6 (F) userType.name
7 (G) roles
8 (H) email
9 (I) password
Normal
Ignore
No problems
2 Timb local Tim Bilder Normal Physicist 4 timb@conaptic.com timbli
3 Tima local Tim Arnold Normal Physicist 4 tima@conaptic.com tim
Import Result Detail
Level Line Column Message
Info -1 -1 Import not saved: found a total of 2 problems
Error 2 -1 Could not create user: Login name 'Timb' is not available
Error 3 -1 Could not create user: Your password must be at least 6 characters long
OK
Import Result
Import not saved: found a total of 2 problems
OK
Show Log

If there are errors a pop up will appear, use the **Show Log** button to show a complete list of the errors. Records with an error will have their record number highlighted in red. Moving the cursor over a record number with an error will result in a tool tip coming up describing the error. The error log just shows the records that failed to import and a reason for that failure. If there are two violation reasons that will cause a record to fail, only one is found, and when fixed, the second reason will cause a failure next time around. This is because the *Time Templates*, *Booking Rules* and *Permissions* are run until one fails and the rest are then ignored.

In the example above the third record has both a password that is too short and an illegal email address, but only the short password has been found, the illegal email address has been found for the second record but not the third. So after fixing the password for the third record and the email address for the second record, if an attempt was made to import the file again with no other changes, another error would occur this time an invalid email for the third record.

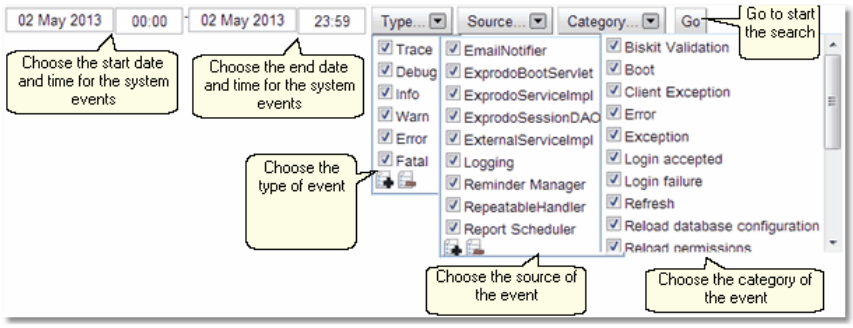
In order to fix these types of errors go back to the original import document and fix them there before loading it back into the **Import** page. Remember once the file to be imported has been updated, save it before reloading the file into **Calpendo Import**.

© 2010-2015 Exprodo Software

4.8 System Events

The **System Events** page provides a convenient way to search for and display **Calpendo**'s record of *events*[449] such as people logging in or emails being sent. It is recommended that only those who administer **Calpendo** should have this added to their menu. To access the **Systems Events** page it is found on the **Admin** menu. However, the administrator may have configured **Calpendo** so that the menu is different.

The **Systems Events** page is very similar to the Search[108] page. The *events* returned can be filtered by time, type, source and category.Unlike other Search[108] pages there is no auto run option. When the page opens up it looks like the following:



Once the search has run the results will look similar to below, clicking on an *event* will bring up the expanded view for that *event* below the list view.

When	Source	Category	Type	User	Affected Type	Affected ID
2 May 2013 00:17	RepeatableHandler	Update	Info			
2 May 2013 02:00	Logging	Error	Error			
2 May 2013 02:00	Reminder Manager	Reminder unsent	Error		Booking	8374
2 May 2013 03:17	RepeatableHandler	Update	Info			
2 May 2013 07:00	Logging	Error	Error			
2 May 2013 07:00	Reminder Manager	Reminder unsent	Error		Booking	7694
2 May 2013 07:17	RepeatableHandler	Update	Info			
2 May 2013 08:17	RepeatableHandler	Update	Info			
2 May 2013 09:10	Logging	Exception	Error			
2 May 2013 09:10	Logging	Exception	Error			
2 May 2013 09:10	Logging	Exception	Error			
2 May 2013 09:10	Logging	Exception	Error			
2 May 2013 09:15	Logging	Exception	Error			
2 May 2013 09:15	Logging	Exception	Error			
2 May 2013 09:17	RepeatableHandler	Update	Info			
2 May 2013 09:20	Logging	Exception	Error			
2 May 2013 09:26	Logging	Exception	Error			
2 May 2013 09:26	Logging	Exception	Error			

When	2 May 2013 08:17
Source	RepeatableHandler
Category	Update
Type	Info
User	
Affected Type	
Affected ID	
Message	2 objects affected, yielding 4 changes

Detail

Type	Original PK	Repeat Type	Affected PK	Action	Range
Template	10	Weekly	16176	Update	2013-05-02 08:00 - 2013-05-02 13:00
			10	Create	2013-05-09 08:00 - 2999-12-31 13:00
Booking	7694	Weekly	9049	Update	2013-05-02 08:00 - 2013-05-02 10:30
			7694	Create	2013-05-16 08:00 - 2999-12-31 10:30

If there are any questions about *system events* please get in touch with **Calpendo** customer support (info@calpendo.com).

© 2010-2015 Exprodo Software

Part V

5 Calpendo Quick Start Configuration Guide

This guide will show a new administrator how to set up **Calpendo** quickly, getting it running with basic functionality.

Resources

Start off by creating each of the bookable *resources*[449] required, such as equipment and rooms etc. To do this go to **Admin->Resource Editor** and click on **Resource**. Then, for each *resource* to be created:

1. Press the **Create** button.
2. Give it a name.
3. Leave the location as **Please select a location**. Locations can be added later.
4. Use one of the existing *resource types*[449] even if they don't quite match what's needed. New *resource types* can be added later.
5. Specify whether a *project*[448] is required to book the *resource*.
6. Leave everything else as default values.
7. Press the **Save** button to save your *resource*.

See Configuring Resources and Locations in the Calpendo documentation.
Once all the required *resources* have been created, they now need to be added to the **Show All** *bookmark*[447] so that the *resources* will appear in the *calendar*[446]. Go to **Admin->Bookmark Manager** and:

1. Under **System Bookmarks** click the **Show All** *bookmark*.
2. Press the **Edit** button to go into edit mode.
3. Press the **Choose Resources** button.
4. Move all the resources from **Available Resources** to **Selected Resources** by selecting each one and pressing the right arrow.
5. When finished press **OK**.
6. Press the **Save** button.

See the Bookmark Manger chapter in the Calpendo documentation.

Projects

Go to **Projects->Create Project** and create a few *projects*. Information stored in a *project* may need to be changed by adding your own *properties*[448], but just use the defaults for now. Then for each *project* to be created:

1. At least give each project a *Project Code*[448], a **Name** and make sure to change the *Status*[448] to **Approved** so that non-administrators can use them for *booking*[446].
2. Go to the **Project Resource Settings** tab.
3. Press the **Choose Resources** button.
4. Move the *resources* that can be booked for this *project* from **Available Resources** to **Selected Resources** by selecting each one and pressing the right arrow.

© 2010-2015 Exprodo Software

5. When finished press **OK**.
6. Press the **Submit project request** button.

See Creating Projects chapter in the Calpendo documentation.

Email

If **Calpendo** is hosted by **Exprodo Software**, then the email will already be configured. Otherwise, go to **Admin->Global Preferences** and configure the email tab as described in *Global Preferences* 447 Email chapter in the Calpendo documentation.

Users

Any new users should be sent a link to the **Calpendo Login** page, and told to use the **Register New User** button to make a user request.

If email is set up correctly, the administrator should get an email telling them about each new user request. To approve a user, go to **Admin->User Request**. For each user to be approved:

1. Click on the **Login Name**.
2. Set **Status** to **Normal.**
3. Set the **Roles** to be assigned to the user.
4. In the **Projects** Tab at the bottom click on **Please Select a Project to add** and select the *projects* to be added to this user. (A user needs to be associated with a *project* before they can use it to make a *booking*).
5. Press the **Save** button.

See the The User Approval Process 153 and The User Requests Page 153 chapters in the Calpendo documentation.

If the users are going to be created by the administrator, go to **User->Search**, press the **Create** button to create a new user. Remember to allocate any *projects* to that user. At least one user should be set up with the **Admin** *role* 450. It is advisable to leave the initial **Admin** and **Root** users alone as this allows **Exprodo Software** access to your **Calpendo** in the event of any problems. For more help read the Modifying Users 163 chapter.

Next Steps

Once the above configuration is done, **Calpendo** will be ready for use.

Calpendo is now working, but there are many additional features or configuration options that are available. To see what else could be configured, see the Before Going Live section of the Initial Configuration chapter in the Calpendo documentation.

It would also be advisable to read through the **Calpendo Users Guide** chapters on the Bookings Calendar in order to get a full understanding on how the *calendar* 446 works and the chapters on The Booking Approval Process 138 and The Booking Request Page 139 in the Calpendo documentation in order to understand how to approve *bookings*.

If there are any questions, please email support@exprodo.com.

© 2010-2015 Exprodo Software

Part VI

6 Calpendo Configuration Guide

Calpendo was designed from the very beginning to be flexible, so that it could meet the requirements of many different types of facilities. It's targeted at people who have shared *resources*[449] that need *booking*[446], but within that, there are many different ways you might like a *booking* system to work.

The **Calpendo Configuration Guide** describes how to change your **Calpendo** so that works the way you want it to. This does not normally need to be read by non-administrators, nor by anybody who only needs to do the day-to-day administration. This guide should be read in conjunction with the **Calpendo User Guide**[28] and **Calpendo Administration Guide**[136].

6.1 Configuration Questions

Calpendo was designed from the very beginning to be flexible, so that it could meet the requirements of many different types of facilities. It's targeted at people who have shared *resources*[449] that need *booking*[446], but within that, there are many different ways you might like a *booking* system to work. Consequently, there are many options for how **Calpendo** can be configured.

When first using **Calpendo**, it is unusual to know exactly (or even approximately) how it is required to be configured for the facility.

This section poses some of the questions to ask yourself as the configurer, but understand that most of them will not be answerable initially. The options are numerous enough that it is expected that the system will be configured over a period of time as a better feel for what can be done is acquired and more information on how it needs to work for the facility is realised. Also, these questions only give a taste of some of the things that can be done.

Please understand that things can be made as simple or as complicated as required. If the facility is relatively new, and the *resources* are not yet fully utilised, then some of the things that **Calpendo** allows may not be required.

Resources

- What *resources* are there that should be bookable?
- Is there going to be a single *resource* to record annual leave?
- Is there going to be a *resource* to represent individual people whose time can be booked?
- Do bookers of a *resource* have to have an allocated *project*[448].

© 2010-2015 Exprodo Software

Booking Requests

- Are users going to create requests that are then approved by an administrator?
- Or do users create pre-approved *bookings*?
- Are *bookings* to be pre-approved for some people and/or some of the time?
- Who can make *bookings* for which *resource*?
- What if any information is required for a *booking* to be made for a particular *resource*?

Booking Restrictions

- Is it required to put users into separate categories for *booking* restrictions, and how might that to work?
- How might the *Time Templates* 449 work? Are *bookings* going to be prohibited in other people's time? Is that behaviour going to change when it gets close to the time? So a slot left free with a day or a week to go becomes up for grabs to everyone?
- Can everyone book as far into the future as they want?
- Are double *bookings* allowed? Perhaps for some *resources* it might be allowed (e.g a computer room with numerous machines)
- Are pre-requisite *bookings* going to be enforced? That is, that a user must book *resource* X before they are allowed to use *resource* Y.
- Is there a limit on how much time a *resource* can be booked for in a given period of time? If so, who should be allowed to book how much time?

Booking Modifications

- Do users have to downgrade approved *bookings* to a request so they can then make changes to it that need to be re-approved?
- Or are users to be allowed to make some changes to *approved bookings* 446?
- Some users may fear cancelling *bookings* because the statistics are collected. They may try to downgrade to a request and then change the time or day (by many weeks) to avoid being seen to cancel. So will users be allowed to change things like the time of day of an *approved booking*, but not allowed the date to be changed.

© 2010-2015 Exprodo Software

Administration

- Who is going to administrate **Calpendo**?
- Who will install and upgrade **Calpendo**?
- Are there going to be separate administrators so some people only see the most common and easier admin options? Should a menu be configured so that a simple **Admin** menu offers the ability to handle *booking* requests, new user requests and *project* requests, but not more advanced administration option, and an advanced **Admin** menu that also gives access to automated emails, *Permissions*, *Time Templates* and *Booking Rules*.

Projects

- What process are required to be put in place for creating new *projects*?
- Will there be a single-step approval process, or a multi-step approval process in which many people each give their approval. Will people need to address questions about whether the *project* is sensible, has ethics approval or has its finances properly arranged.
- What questions need to be asked of users when they create a new *project* request? For example, there might be safety related questions about what they are intending to do.
- What information needs to be stored on a *project*?
- What per-*project* and per-*resource* information do needs to be stored? For example, the price for using each *resource* for a *project*.

Services

- What services are available?
- When ordering a service what information is required?
- Costing for services.

Automatic Emails

- Are emails to be sent when *bookings* are cancelled?
- Should any cancellation trigger an email, or just some of them?
- Who should receive such emails?

Permissions

- Who should be allowed to do what within the system?

© 2010-2015 Exprodo Software

6.2 Initial Configuration

When **Calpendo** is first installed, have a play with it to help understand the way it works, and to become more familiar with the configuration options. There are some things that need to be set up as soon as you start playing with **Calpendo**, and some things that must be done before going live and using **Calpendo** for real. These are described here.

Resources

For each *resource* 449 that is to be bookable (rooms, scanners, people or whatever), create a *resource* using the **Resource Editor** as described by Resources 270.There is no need to worry about *resource groups* 449 or locations just yet. Just make sure that all the *resources* required for now are created.

Email Preferences

Tell **Calpendo** how it should send email, so that Automatic Emails, reminders and manual emails will work properly. Set the email preferences on the Email tab of the Global Preferences 328 page.

If **Calpendo** is hosted on **calpendo.com**, then the SMTP host and emailed base URL will have been configured already. However, it is possible to choose the name that **Calpendo** reminders and Automatic Email appear to come from.

New User Requests

When a new user registers in **Calpendo**, by default **Calpendo** is configured to send an email to administrators. This means any user registered in the system whose roles include the **Admin** role. Who gets Automatic Emails 297 can be changed, but for now it is best to make sure that there is at least one user with the **Admin** role that has a valid email address configured. See Modifying Users 163 for information on how to do that.

Before Going Live

These are the things that need to be done before you go live, and use **Calpendo** for real:

- Work out what *properties* 448 should exist on a Project 205 and *bookings* 446 for different *resources*.
- Work out how *projects* 448 will be approved, and who will assign the *project* a unique *code* 448.
- Use the Bakery 351 to configure the *properties* on a Project 205 and then modify the project template 206, which is copied each time a new *project* is created.
- Use the *Bakery* to create any required **Booking** sub types and configure their *properties*.
- Create one or two *projects* in **Calpendo**, and configure them to mirror real *projects* to see if all the information required can be specified.
- Work out whether to assign all users a *User Type* 450, and whether to assign all projects a *Project Type* 448. It's much easier to assign the type to users and *projects* as

© 2010-2015 Exprodo Software

they are created, so it's better to work out how this is to work before creating users and *projects*. See Configuring Types And Groups 266.

- Identify who will perform day-to-day administration of the system. This means working out who will approve *bookings* or *projects* (if appropriate) and who will users go to if they can't make a *booking* (because **Calpendo** has been configured to limit what they can do). Also, while **Calpendo** will make the management of the shared *resources* much easier, there are always some problems that require intervention to fix. For example, if some equipment breaks down.
- Customise the FAQ 325 with questions pertinent to your installation especially who to contact in case of problems.
- Configure user *roles* using the bakery, if you want to add more *roles* to the system. These can be used by Booking Rules 223, Permissions 278, Automatic Emails 297 and Manual Emails 172
- Approve new users 153 as they register or change the roles 163 bestowed upon users that have already registered.
- Finish resource configuration 270 by choosing **Booking** sub types, locations, types and groups and choosing whether *bookings* for each *resource* require a *project*.
- Set up any Automatic Emails 297 so that people receive notification about events.
- Set up Permissions 278 to control who can do what, with what and when.
- Create Time Templates 215 that earmark time for some users or *projects*.
- Create Booking Rules 223 to control who can make what *bookings*. For example, to limit double *bookings* or the amount of time that can be booked by each user or *project*.
- Create custom searches 108 and reports 121, such as *project usage* 449 of *resources*, billing etc.
- Customise menus with the Menu Editor 311 and by assigning menus to users 167 or setting the default menus for each user role 341.
- Create or modify the Frequently Asked Questions 325 that are shown to the users.
- Configure the global preferences 328.
- Finally, define Dynamic *Biskits* 446 using the *Bakery* in **Calpendo** to be able to create, read, update and delete any other table required, in the **Calpendo** database, subject to some limitations. This may be required if actual *resource usage* is captured (as opposed to *bookings*), and this needs to be stored in the **Calpendo** database and **Calpendo** used to examine and compare this with *bookings*. Alternatively, use **Calpendo** as a general-purpose data editing engine.

© 2010-2015 Exprodo Software

6.3 More Calpendo Configuration

Once **Calpendo** has been configured and it has been used for a while, there may be other functionality that you would like it to provide. Here are some possible examples.

- Do you need to record different information when booking different *resources*[449]*?* Set up a **Booking Sub Type** *Biskit*[446] so different *resources* have different *properties*[448] to be filled in when being booked. See the section on Setting Up Different Booking Sub Types For Resources[203] below.
- Do you want to attach documents to some *Biskits*[446]*?* To do this, set up a *property* or set of *properties* (for more than one document) to hold file attachments. See the section on Attaching Documents To Biskits[204] below.
- Do the users need training before using a *resource?* One way to do this is to set up a **Training** *Biskit Type* in the Bakery[351] and use the **Search Results Booking Rule** to make sure a *booking* cannot be made without the appropriate training record. See the section on Search Results Booking Rule[250] for ideas on how this might work.

Setting Up Different Booking Sub Types For Resources

In order for *resources* to have different information on their booking forms, it is necessary to set up different **Booking** *Biskit Types* for those *resources,* known as **Booking Sub Types**.

In order to do this first create a *Biskit Type* that inherits from **Booking** (see Creating An Inheriting BiskitDef[412]) and then add the *properties* that are required for that *Biskit Type* (see Adding Properties[384]).

Once a **Booking Sub Type** has been defined in the *Bakery*, the following will be enabled:

- The **Resource Editor** allows selection of which **Booking Sub Type** should be used for its bookings (see Configuring Resources And Locations[270]).
- The **Booking Rule Editor** allows selection of the **Booking Sub Type** that it applies to. Any *Rule*[446] that applies to *bookings* of type **Booking** will also apply to any *bookings* of the **Booking Sub Type**. A *Rule* that applies to a **Booking Sub Type**, will not apply to its parent i.e. **Booking** (see Choosing Which Bookings A Rule Applies To[226]).
- **Booking** searches allows selection of the **Booking Sub Type** to be searched for. Search for **Booking** and all **Booking Sub Types** will be found as well. However,in order to filter the returned *bookings* by a *property* unique to a **Booking Sub Type**, then make the search look only for that **Booking Sub Type**.
- A *Permission*[448] that applies to **Booking** will also apply to any **Booking Sub Type**, although the reverse does not apply. See How Permissions Work[278] for details on how *permissions* apply to sub types.

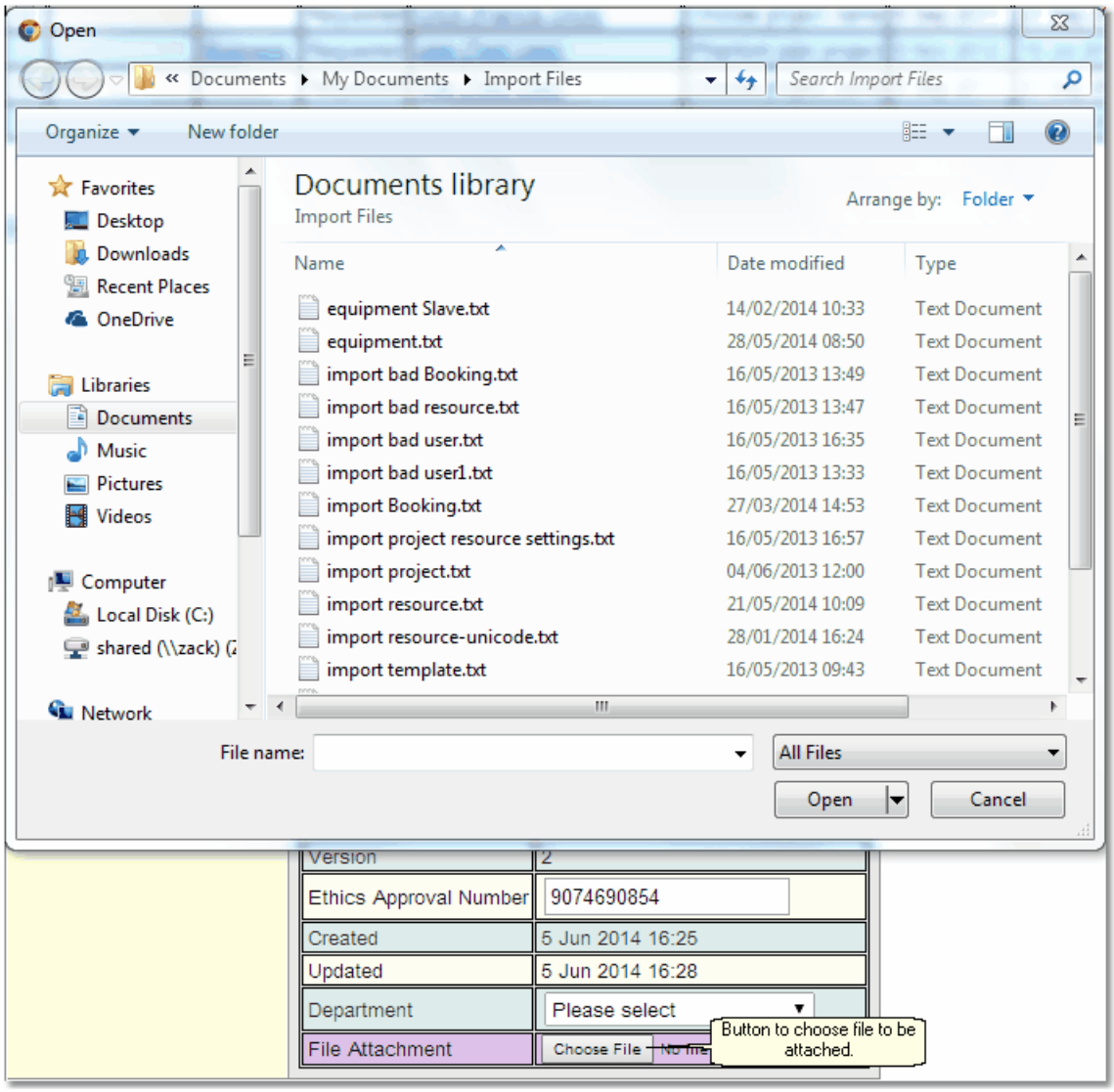
If an administrator wishes to move to **Booking Sub Types** for *resources* with existing *bookings,* then they need to talk to the **Calpendo Support Team** in order to help with the transition including moving across historical data.

© 2010-2015 Exprodo Software

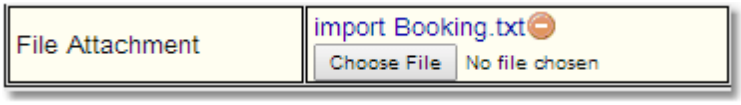
Attaching Documents To Biskits

In order to attach documents to a *Biskit* there needs to be a *property* created on its *Biskit Type* that will hold that document (see Adding Properties For File Attachments 390). If a number of documents need to be attached to a *Biskit,* then a number of *properties* could be used or a single *property* of type **Set** (see Creating A Set Of BiskitDef 398).

Once the *property* has been added, edit the *Biskit*, and there will be a **Choose File** button, which when pressed will provide a file browser so the file to be attached may be chosen.



Once selected, the file name is shown along with a (red) button to remove the attachment. Pressing **Choose File** will replace the attached file with a new one.



© 2010-2015 Exprodo Software

6.4 Converting From An Existing Booking System

If there are already *bookings* [446] and possibly *projects* [448] that you would like **Calpendo** to take over, then the process of migrating to **Calpendo** may feel daunting. The following method is the recommended route:

- Pick a date in the future, maybe 6 to 8 weeks away.
- Any *bookings* for the *resources* [449] beyond that date should be made through **Calpendo**. Any existing *bookings* should be imported into **Calpendo**. This can be done manually, and by picking a date far enough into the future, it should reduce the number of *bookings* sufficiently so that a manual import process is acceptable. If there are too many *bookings* to do this manually, and the existing *bookings* are stored in an electronic system, then it may be possible to automate their import. Email info@calpendo.com for help with this.
- If there are existing *projects*, then create them in **Calpendo**. It is important that the *properties* [448] on a *project* get configured to your requirements first, so that when the *projects* are created, all the information is provided.
- Send an email to everyone who may want to use **Calpendo**, and invite them to register. If everybody is getting a *user type* [450], then it is important to configure the user types [266] before sending this email. This is because when users register, they select their own *user type*. An administrator can check or change the *user type* if required.
- Associate all users with their *projects*. This should be done after you've created the *projects*, and after most of the users have registered. It needs to be done before users start making their own *bookings*. This is because users can choose a *project* they are associated with when making a *booking*. Some or all *resources* can be configured to not require a *project* with their *bookings*.

There may be some apprehension about the process of switching to using **Calpendo**, however those that have converted so far have been pleasantly surprised about how painless it was.

Also, the above is only a suggestion and other methods may well work. For example, if there are multiple *resources* to migrate, it may make sense to migrate them one at a time.

6.5 Project Configuration

One of the most important aspects of configuring **Calpendo** is to make sure that *projects* [448] are set up the way you want. This means making sure that the *projects* store the information required for the facility and that *projects* are approved using a suitable mechanism.

See Also

- Types And Groups [265] to understand what types and groups are, and what the difference between them is.
- Configuring Types & Groups [266] to understand how to go about setting them up for *projects*, as well as for users and *resources* [449].

© 2010-2015 Exprodo Software

6.5.1 Project Template

Whenever somebody creates a new project[70], **Calpendo** first creates a copy of a *project*[448], referred to as the template *project*. It doesn't have to be a special *project* that is used as the template, but it's generally appropriate that the template *project* is not used for *booking*[446] *resources*[449], since real *projects* usually have lots of information stored in them which is not on the template. For this reason, the template *project* is normally assigned a status of **Unbookable** so that nobody can create a *booking* that references it.

Choose which *project* is used as a template using the Global Preferences[343] setting for the template project[343].

Note that, although *properties*[448] from the template *project* are copied when creating a new *project*, some *properties* are modified. These are:

Property	Description
Name	This is always reset to **Unique project name**.
Owner	This is changed to the current user.
Users	The current user is *added* to the list of users.
Status	The status is changed to **Requested**.
Project Code	This is set to be empty.

The template *project* ignores the need for values to be input for all *properties* that are set up to have required values.

6.5.2 Configuring Project Properties

It seems that everybody has a different idea about what information should be collected in a *project*[448]. For example, some facilities will need to collect safety information that is particular to the *resources*[449] they have; if there is an MRI scanner, you will probably want to know what people are intending to do both in the scanner room and possibly outside. Are they going to use lasers, gases, pain devices, collect blood or give drugs? Are they going to take any new equipment into the scanner room, and if so, what? These questions are just examples of things that may be appropriate for one facility and not for others. This is why **Calpendo** allows you to configure your *projects* to contain the *properties*[448] that make sense for your facility.

© 2010-2015 Exprodo Software

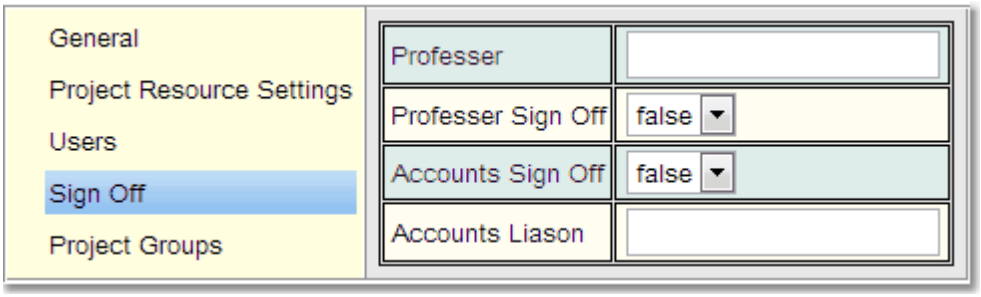
Some *properties* exist for every **Calpendo** installation, and these are described in Project Properties[144]. For example, every project has a name, status and a list of users that are associated with the *project*.

The *properties* on a new *project* in a new **Calpendo** installation are a default set that may well not be appropriate for your use, and so as well as adding your own required *properties*, it may also be necessary to remove some.

Adding, removing and relabelling *properties* is done in the Bakery[351]. *Properties* can be added to *bookings*[446] as well as *projects*, so its possible to always capture the information required.

When creating your own *properties* for *projects* each *property* has a 'required' attribute that can be selected, this will force a user who is creating a new *project* to enter data into this field. For text or numerical properties there can also be specified a minimum and/or maximum value. For text *properties*, the minimum and maximum refer to the length of the text provided by the user. This means a user could be required to provide at least 50 characters (or whatever) to describe their *project*.

If a large number of *properties* are being added to the **Project** *Biskit*[446] and they need to be displayed in particular groups under specific tab names then use the **Group** *meta-property*[448]. All *properties* with the same **Group** *meta-property* will be displayed under the same tab which will be labelled with the value of the **Group** *meta-property*.



Permissions[278] can be used to control who can enter what for a *project*. For example, each *project* has a unique *project code*[448], and it may need to be limited as to who can change that *property*. By setting the *Permissions*[448] in such a way that only the administrators may set the *project code* on a *project*, a user will not even be given the opportunity to enter a *project code* when creating or editing a *project*.

Additional *properties* may also be required if there is going to be a multi-step approval process. This is covered by the next section, Configuring The Project Approval Process[208].

© 2010-2015 Exprodo Software

6.5.3 Configuring The Project Approval Process

This section describes how to set up the *project approval process* [448] that is required. Please make sure to read The Project Approval Process [145] first, which describes some of the options and how the *approvals process* works from the perspective of the person that does the day-to-day administration of **Calpendo**.

As described in The Project Approval Process [145], there are two approaches to project approval: the single-step approval or the multi-step approval. This section will not cover the question of why to choose one method over the other, but will instead focus on how to set up each method.

Final Approval: When A Booking Can Use A Project

A *project* [448] can be associated with a *booking* [446] when its *status* [448] is set to **Approved**. This means that changing a *project's status* to **Approved** is the final step of a multi-step *approval process*, and the only step in a single-step *approval process*. The approval itself (that is, changing the *status*) is described in The Project Requests Page [146].

Single-Step Approval Process

As mentioned above, if a single-step *approval process* is being used, then the *project status* just needs to be set to approve or deny. Nothing special in this case needs to be configured. However, it is possible that **Calpendo** has been pre-configured with some *properties* [448] intended for use in a multi-step *approval process*. Therefore, there may be some *properties* on *projects* that are not required. See Configuring Project Properties [206] for information on changing the *properties* on a *project.*

Multi-Step Approval Process

A multi-step *approval process* does not have specific support in **Calpendo**. However, the effect can be achieved by adding properties to **Project** to store the various approval stages. For example, suppose it is required to provide separate approvals for the science, finance and ethics before final approval can be given. In this case, the recommended configuration is as follows:

1. Create boolean (Yes/No) *properties* on **Project** called **scienceApproval**, **financeApproval** and **ethicsApproval** using the Bakery [351] as described by Configuring Project Properties [206].

2. Set up Permissions [278] so that only the desired people can set **scienceApproval**, **financeApproval** and **ethicsApproval** to true on any *project*.

3. Set up *Permissions* [448] so that only the desired people can modify the *project status* to **Approved** and only then when **scienceApproval**, **financeApproval** and **ethicsApproval** are already set to **True**.

4. Set up *Permissions* so that only the desired people can modify the *project status* to **Denied** but that this can be done regardless of **scienceApproval**, **financeApproval** and **ethicsApproval**.

© 2010-2015 Exprodo Software

If the approvals need to be given in a particular order, then configure this using the *Permissions*, so that whoever gives the financial approval may only do so once the ethics approval has been given, for example.

It would be a good idea to also set up Automatic Emails 297 to help notify people when it's time for them to give their approval. To do this, make sure that there are a set of *conditions* 447 on the Automatic Email so that it is triggered by a change that means the *project* is now in a state suitable for them to give their approval.

© 2010-2015 Exprodo Software

6.6 Controlling Bookings

Calpendo offers three ways to control *bookings*[446]. This section describes how to choose between the three, Time Templates[215], Booking Rules[223] and Permissions[278]. It also describes *Time Templates*[450] and *Booking Rules*[446].

The Automatic Booking Approval Process

First some preferences that are set up globally that will affect the process**:**

Default *booking status*[447] set up in Global Preferences->Bookings->Default Booking Status[331],

1. Regular users the default value is **Requested**.
2. **Admin** users the default value is **Approved**

Also by using Global Preferences[328] it is possible to specify that Time Templates[349] and Booking Rules[345] do not apply to **Admin** users.

The process:

1. When a user creates a *booking*, they get a *booking* pop up with the *booking status* set to **Best Possible**.
2. The user may change the default *booking status* for that *booking* if they have permission[448] to do so.
3. Once the user presses the **Create Booking** button, the **Calpendo** Client will check the *booking* against any Time Template[449] that exists for that time period and resource[449]:
   a. If the *booking* is automatically approved the *booking status* is set to **Approved** and the property[448] **Approved by Template** is set to *True*.
   b. If not **Automatically Denied** it will keep the *booking status* chosen in the pop up when the *booking* was created.

The full list of possible outcomes is in the table below (if there is no outcome mentioned, that combination of chosen *booking status* and **Default Booking Status** are not possible):

Template Response	Initial Chosen Booking Status	Final Booking Status Default Booking Status = Requested	Final Booking Status Default Booking Status = Approved
No Template	Best Possible	Requested	Approved
	Requested	Requested	Requested
	Approved		Approved
Warning Template	Best Possible	Requested	Requested
	Requested	Requested	Requested
	Approved		Approved by Template

© 2010-2015 Exprodo Software

Template Response	Initial Chosen Booking Status	Final Booking Status Default Booking Status = Requested	Final Booking Status Default Booking Status = Approved
Acceptable	Best Possible	Requested	Requested
	Requested	Requested	Requested
	Approved		Approved by Template
Auto-Approval	Best Possible	Approved by Template	Approved by Template
	Requested	Approved by Template	Approved by Template
	Approved		Approved by Template
Auto-Denial	Any	Denied	Denied

If the *booking* is not automatically denied it is then passed to the server and depending on the current *booking status* (and **Approved by Template** *property*) more checks are done:

a. **Requested**:

i. the server checks against the *Time Template* again and any messages associated with that *Time Template's* target group is displayed

ii. the *Booking Rules* are checked, and if the *booking* is denied a customised message may then be displayed

iii. the *Permissions* are checked, and if the *booking* is denied the **Permission Denied** message will then be displayed

iv. The **Calpendo** administrator will now have to manually approve the *booking*.

b. **Approved by Template**:

i. the server checks against the *Time Template* again and any messages associated with that *Time Template's* target group is displayed

ii. the *Booking Rules* are checked, and if the *booking* is denied a customised message may then be displayed

iii. the *Permissions* are checked, and if the *booking* is denied the **Permission Denied** message will then be displayed

c. **Approved**, (only for Users with a **Default Booking Status** of **Approved** who do not have a *Time Template* applying to them)

i. the *Booking Rules* are checked, and if the *booking* is denied a customised message may then be displayed (this is optional as **Admin** users may be set up to not have *Booking Rules* applying to them)

ii. *Permissions* are checked, and if the *booking* is denied the **Permission Denied** message will then be displayed

The *booking* is now approved.

See Also:

Permissions 278

© 2010-2015 Exprodo Software

6.6.1 Choosing Between Time Templates, Booking Rules and Permissions

Calpendo allows control of the *bookings* [446] that people are allowed to make using Time Templates [215], Booking Rules [223] and Permissions [278]. Some controls to be implemented could be done using any of these three, so it's important to understand the advantages and abilities of each method so that the most appropriate methodology is used to achieve what is required.

The general rule of thumb is to preferably implement controls using *Time Templates* [450] where possible. This is because the *Time Templates* can be displayed on the background of the *Bookings Calendar* [446] so that users will be able to see at a glance when they will be allowed to make a *booking*. However, *Time Templates* only allows the placement of controls on the times when *bookings* can be made, whereas *Booking Rules* [446] and *Permissions* [448] offer control in other ways.

Let's look at each option with an example that can be implemented in all three ways. Suppose it is necessary to stop a particular user from making *bookings* on Mondays. Here is an outline of how to implement this in each methodology:

Time Templates Example

Use the **Time Template Editor** to create a *Time Template* that applies to the person (or people) to be controlled, and set the *bookings* acceptability to **Automatic Denial**. Also specify a message that would be given to the user if they tried to make a *booking* in the forbidden time. Once the *Time Template* is created, then choose the times to apply it. For this, go to the *Time Templates* page and create a *repeating* [449] entry, and apply the *Time Template* every Monday.

When the user logged in, they would then see the background of every Monday in the *bookings calendar* in red, and letting the mouse hover over the red background would show them the message that has been entered. If they tried to create a *booking* on a Monday, then as soon as they click on a Monday, they would be given the error message. This means *Time Templates* can give instant feedback to the user, before they even enter the details of the *booking*.

Booking Rules Example

To stop somebody booking on Mondays using *Booking Rules*, use the **Booking Rule Editor** [223] to create a Simple Booking Rule [230]. Specify that the *Booking Rule* applies to the person or people you want to control, using the **Bookers** tab within the **Applies To** tab. In the **Conditions** tab, specify a *condition* [447] that the *booking's* start day is a Monday. Next, specify the *Booking Rule's* **Rejection Type** as **Reject** and enter a message to be shown to the user.

When the user logged in, they would now see no indication that they cannot make a *booking* on a Monday. Only after entering the details of a *booking* they wanted to make would they see the error message that was set up in the *Booking Rule*.

© 2010-2015 Exprodo Software

Permissions Example

To deny somebody *permission booking* on Mondays, you would create a *Permission* that applied to *Bookings*, add a *Condition* the same as for *Booking Rules* that the *booking's* start day is a Monday, and use the **Applies To** tab to specify the users that the *Permission* *applies to* 446.

When a user tries to make a *booking* that is denied by a *Permission*, then they would see an error message that says **Permission Denied**. There is no way to provide your own error message to be shown when using *Permissions*. Consequently, this is the least user-friendly way of controlling *bookings* and so should be used sparingly.

From reading the examples above, it may appear that there's not a huge difference between the three methods. However, they differ in the information that they can each use, and in the results they can effect:

Time Templates: Information In And Out

Time Templates easily allows a choice of times when bookings should be requests, automatically approved, given a warning or automatically denied. They also allow decisions based on who makes the *booking*, the *project* 448 it's for, and the *resource* 449 it's for. No other information can be used by *Time Templates*.

Booking Rules: Information In And Out

Booking Rules can be configured to use *any* information. There are different types of *Booking Rules* that make it relatively easy to apply some controls, for example to prevent double *bookings* or limit how much time somebody can book in a week. However, a *Booking Rule* can be made to do almost anything by creating an Advanced Booking Rule 256 (although this requires a programmer to write some Java code).

Permissions: Information In And Out

Permissions can only use the information in the *booking's* *properties* 448, as well as information about who was making the *booking* and when. There is no access to other information in the **Calpendo** database. For example, *Permissions* cannot be used to control the time booked per week or to prevent double *bookings* because that would require information about other *bookings*.

© 2010-2015 Exprodo Software

Comparing Time Templates, Booking Rules And Permissions

Comparing the methods gives us these differences:

- They each provide different levels of feedback to the user. *Time Templates* provide graphical feedback before a user tries to make a *booking*, and can stop a user from even entering a *booking's* details by providing a message you have specified in the *Time Template*. *Booking Rules* provide no graphical feedback and can only give an error message after the *booking* details have been entered, but the message that will be provided can be chosen. *Permissions* also provide no graphical feedback, and can only give a generic **Permission Denied** error message after the *booking's* details have been entered.
- A single *Time Template* can provide a different message to different groups of people. *Booking Rules* and *Permissions* can each only apply to one set of people.
- *Time Templates* can be used to choose whether, when and which *bookings* should be made as a request or automatically approved. *Booking Rules* and *Permissions* can only deny or accept a booking. For example, *Booking Rules* and *Permissions* cannot choose to automatically approve a *booking*.
- *Time Templates* are the most convenient way to provide a warning for a specific period of time. For example, if a user is making a *booking* for an MRI scanner, and there are no operators available one particular week due to annual leave, there could be a *Time Template* that applies during that week to give users a warning that they need to make their own arrangements for a scanner operator to be available. While a warning message can be provided using *Booking Rules*, it's not as flexible in setting the start and finish time it should apply, and it provides no feedback to the user until after they make their *booking*.
- *Time Templates* work when the control to be applied relates to when the *booking* is made. *Permissions* apply to any *property* of the *booking*, while *Booking Rules* can use any information even if it's not on the *booking*. For example, *Time Templates* and *Permissions* cannot be used to prevent double *bookings*, or to limit the amount of time somebody can book per week.
- *Time Templates* are simpler than *Booking Rules* and *Permissions*.
- Permissions 278 are the most difficult of all three techniques to understand. This is because many *Permissions* may be set up, some of which give *permission* and some of which may refuse *permission*. **Calpendo** then chooses which *Permission* is the definitive one from all that may match depending on their relative priority. This means they can be confusing because if many of them are created, it's not always obvious which one will be definitive.

In summary:

1. If you can achieve what you want with a *Time Template*, then you should use that.
2. If the control needed only requires information that can be found on the *booking*, or the time the *booking* is being made, and it is not a problem that the user is given a generic **Permission Denied** error message, then it is possible to use a *Permission*.
3. If the above don't apply, then a *Booking Rule* is needed.

© 2010-2015 Exprodo Software

6.6.2 Configuring Time Templates

Time Templates[450] let you assign an acceptability rating to *bookings*[446] depending on when they are booked, who makes the *booking* and the project the *booking* is for. This is done in two separate parts:

1. Create one or more specifications of acceptability according to who makes the *booking* and the *project*[448], but ignoring when the *bookings* are for.
2. Apply those specifications to various periods of time.

The first part is done by the **Time Template Editor** and the second part is done on the *Time Templates Calendar*[450].

6.6.2.1 How Time Templates Work

New Laboratory Example

Calpendo's *Time Templates*[450] which are created in the Time Templates Editor[217] are quite powerful, and best explained by an example. Suppose a new laboratory opens with a scanner, and the building is only open from 7am to 7pm, Monday to Friday. Since the laboratory is new, its scanner doesn't get much use yet, so there is no need to enforce much in the way of *booking*[446] controls, but it is required to prevent people from making out of hours *bookings*. To do this, create two *Time Templates*[449]: one for when the building is open, and another for when it is closed.

The two *Time Templates* are represented by the contents of this table:

Time Template Name	Applies To	Acceptability	Message
Open	Everybody	**Automatic Approval**	
Closed	Everybody	**Automatic Denial**	No *booking* at this time because the building is closed

This means that the **Open** *Time Template* allows everybody to make *bookings*, and those *bookings* will be automatically approved by **Calpendo**. The **Closed** *Time Template* stops everybody from making *bookings*, and whenever somebody tries, they will be given the message shown in the table about the building being closed.

The missing element from all this is to choose the times when the **Open** and ***Closed*** *Time Templates* apply. This is a process much like the Bookings Calendar[36], but instead this is a calendar where you choose the times and *resources*[449] that you want to apply to *Time Templates*. Create an entry on Monday from midnight to 7am that applies the **Closed** *Time Template*, another entry from 7am to 7pm that applies the **Open** *Time Template*, and another entry from 7pm to midnight that applies to **Closed** *Time Template*. In each case, you make them *repeat*[449] weekly for each day Monday to Friday. Finally, create an entry on Saturday and Sunday that applies the **Closed** *Time Template* all day long.

© 2010-2015 Exprodo Software

Exceptions To The Rule

Suppose that the scanner requires trained operators to use it, and that one or more operators are on holiday for a week. Anybody who makes a *booking* during that week will need to ensure that somebody suitably trained will be present. In this case, people are allowed to make *bookings*, but its needed to make sure they know that this problem exists. To handle this, create another *Time Template* that looks like this:

Time Template Name	Applies To	Acceptability	Message
Operator Shortage	Everybody	**Automatic Approval**	There is a shortage of scanner operators at this time. Please make sure operator will be present when you use the scanner.

Then add entries to the *Time Templates Calendar*[450] that applies the **Operator Shortage** *Time Template* for the week in question, from 7am to 7pm each day. Now, when somebody tries to make a *booking* during this week, and during regular opening hours, there are two *Time Templates* that apply: **Open** and **Operator Shortage**. When **Calpendo** sees multiple *Time Templates* that apply, it always enforces the most restrictive one. In this case, both *Time Templates* have the same level of acceptability, but only one has a message attached to it, **Operator Shortage**, and so this is the one that will apply. This means that *bookings* made in this week will still be automatically approved, but will show the message about the shortage of scanner operators.

If required, create the **Operator Shortage** Time Template so that its acceptability is *Warning* instead of *Automatic Approval*. By doing this, two things will change:

1. *Bookings* would be created with a default *booking status*[447] as defined by **Global Preferences->Bookings->Default Booking Status** which by default is **Requested** instead of **Approved** for normal users, so that they should be manually approved[139], and **Approved** for **Admin** users.
2. *Bookings* would record the fact that the user making the *booking* had been warned

© 2010-2015 Exprodo Software

6.6.2.2 The Time Template Editor

The **Time Template Editor** allows the specification of the acceptability of *bookings*[446] according to who makes the *booking* and the *project*[448] it is for, but without worrying about the periods of time when those specifications apply.

The editor is, by default, found on the menu as **Admin-->Template Editor**.

Each item that the editor lets you create consists of the following:

<table>
<tr><th>Property</th><th colspan="2">Description</th></tr>
<tr><td>Name</td><td colspan="2">This is the name used to represent the Time Template[449]. It can be anything required, and will be shown on The Time Templates Calendar[220].</td></tr>
<tr><td>Colour</td><td colspan="2">This indicates the colour that will be used to represent the Time Template when shown on The Time Templates Calendar[220].</td></tr>
<tr><td>Time Template target groups</td><td colspan="2">This is a list of items, each of which identifies a users and projects, and assigns them an acceptability and a message to be shown to the user:</td></tr>
<tr><td></td><td>Property</td><td>Description</td></tr>
<tr><td></td><td>Name</td><td>This is the name used to represent this selection of users and projects.</td></tr>
<tr><td></td><td>Applies To</td><td>This specifies who this Time Template target group applies to.</td></tr>
<tr><td></td><td>Acceptability</td><td>This is where to specify the acceptability for bookings that fall into this Time Template target group.</td></tr>
<tr><td></td><td>Message</td><td>This is a message that should be delivered to the user whenever they attempt to create or update a booking that matches this Time Template target group. If a message is not specified, in the Time Template, and the Time Template's acceptability indicates the user should be given a warning, then they are presented with the default warning message.</td></tr>
</table>

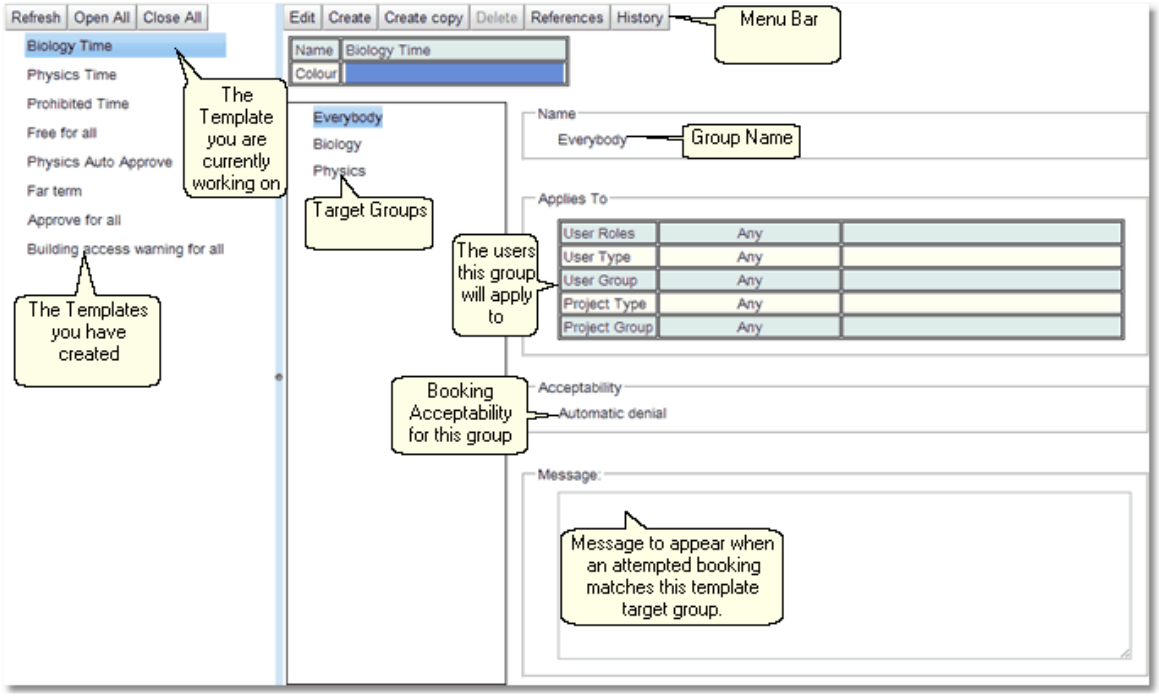
© 2010-2015 Exprodo Software

This table shows what each of the *acceptability* options mean:

Acceptability	Description
Automatic Approval	This indicates that a *booking* will be automatically approved. As such, the *booking status* 447 will be set to **Approved** so that nobody has to manually approve a *booking* request.
Acceptable	Indicates that the *booking* may be created, and it will be given the *status* **Requested**.
Warning	Indicates that the *booking* may be created, and it will be given the *status* requested by the user the same as **Acceptable**, but the user will be warned that something is wrong, or that they may need to do something in particular. For example, it may be required to warn them that there are no operators available and they need to provide their own. Whenever a user is warned, the *booking* records that fact. The only difference between **Acceptable** and **Warning** is that **Warning** records the fact that the user was warned on the *booking*. **Acceptable** and **Automatic Approval** *bookings* may also provide a message to the user if you want.
Automatic Denial	Indicates that the *booking* is not permitted. No *booking* will be created, and an error message should be provided so that it can be shown to the user.

This is what the **Time Template Editor** looks like once opened up.

In the left hand pane is a list of all the *Time Templates* created so far.
In the right hand pane will be the currently selected *Time Template*. The following example is a *Time Template* that is not in edit mode.

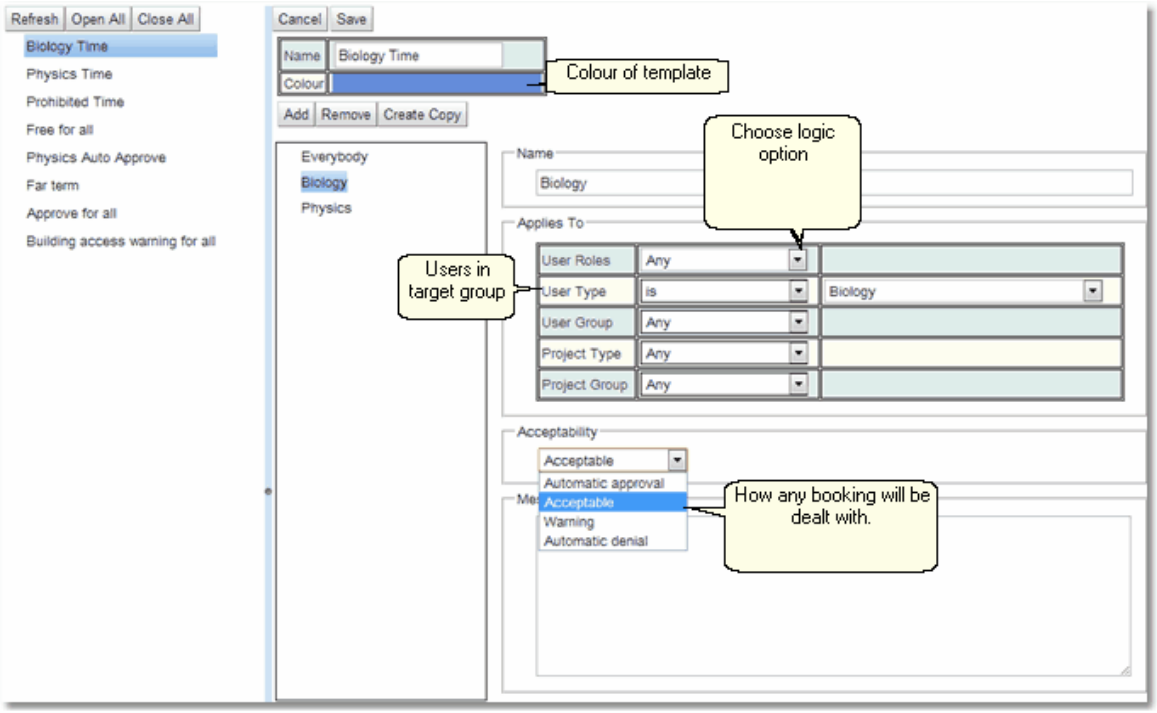


© 2010-2015 Exprodo Software

Each *Time Template* can have a number of target groups, each group will have a list of the users or *projects* that the group will apply to with the *booking* **Acceptability** that will be applied to that group, and a message if an attempted booking has matched the target group.
If an attempted *booking* matches more than one target group within a *Time Template* then the least restrictive group will apply.
If a *booking* matches multiple *Time Templates* then the most restrictive *Time Template* will apply.

For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition 96 chapter.

The following is a *Time Template* in edit mode.



For a target group choose the users to constitute the group. If multiple **Applies to** are set up then any user that complies with all the restrictions is a member of the group. For a user that is in multiple target groups (in the example above, all users will be in the everyone target group, and those with a *User Type* 450 of **Biology** will be in the **Biology** target group) the **Acceptability** that is chosen is the one with the least restrictions. (i.e. in this order **Automatic Approval**, **Warning**, **Automatic Denial**).

Any target group that gets an **Automatic Denial** or **Warning**, should have a message set up, although if there is no user defined message an automatic one will be generated by **Calpendo**.

Once the target groups are set up for a *Time Template*, click on the **Save** button to save the new *Time Template*, or **Cancel** to discard.

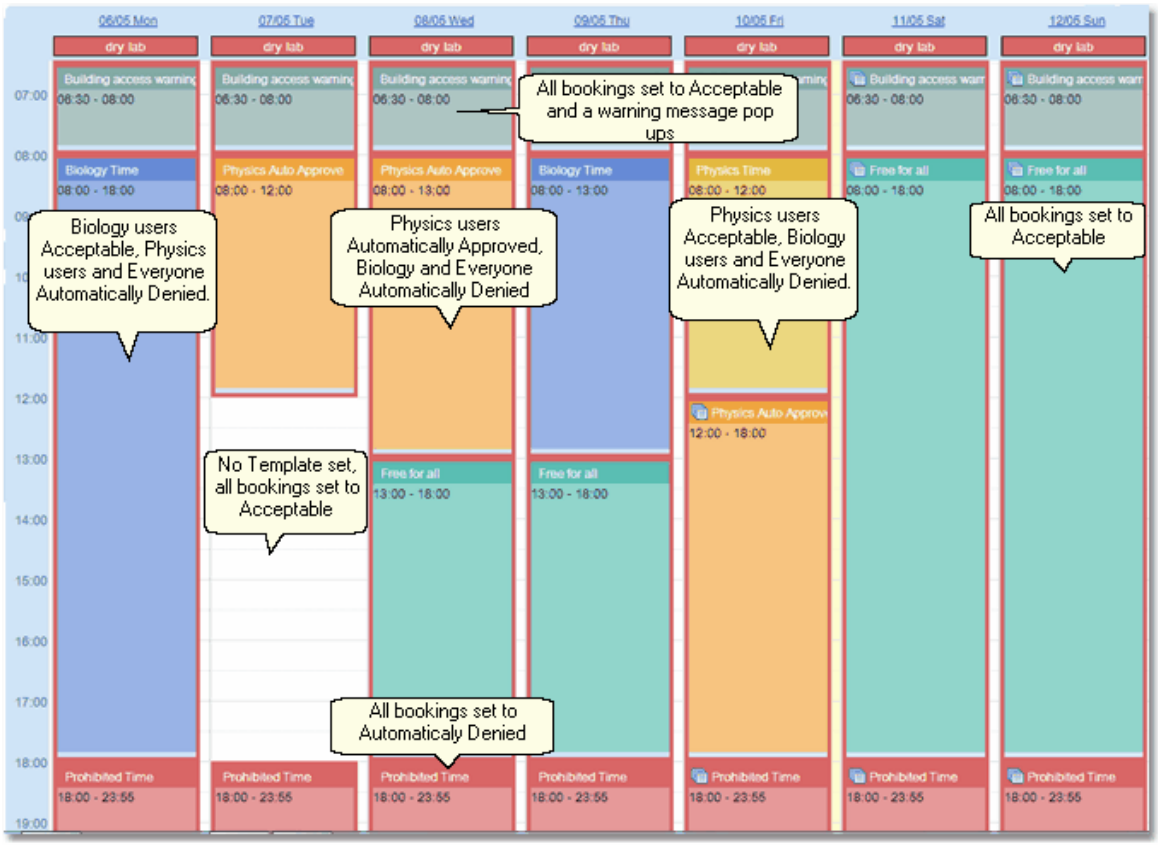
© 2010-2015 Exprodo Software

6.6.2.3 The Time Templates Calendar

The *Time Templates Calendar*[450] works much like the Bookings Calendar[446]. Create a *Time Template*[449] entry on the *calendar*, and specify the time period that it applies for. Also it is possible to set the *Time Template* entry to *repeat*[449]. For example, to earmark Tuesday mornings as **Physics Auto Approve**, then create a *Time Template* entry on a Tuesday morning, set it to use the **Physics Auto Approve** *Time Template* and make it *repeat* every week.

Here is an example of *Time Templates* set up for a *resource*[449] over a single week.
A *Time Template* can choose one of four possible states for a *booking* request:

1. **Automatic Approval**
2. **Acceptable**, *bookings*[446] will be given the status **Requested**.
3. **Warning**, *bookings* will be given the status **Requested** but a warning message will be issued.
4. **Automatic Denial**

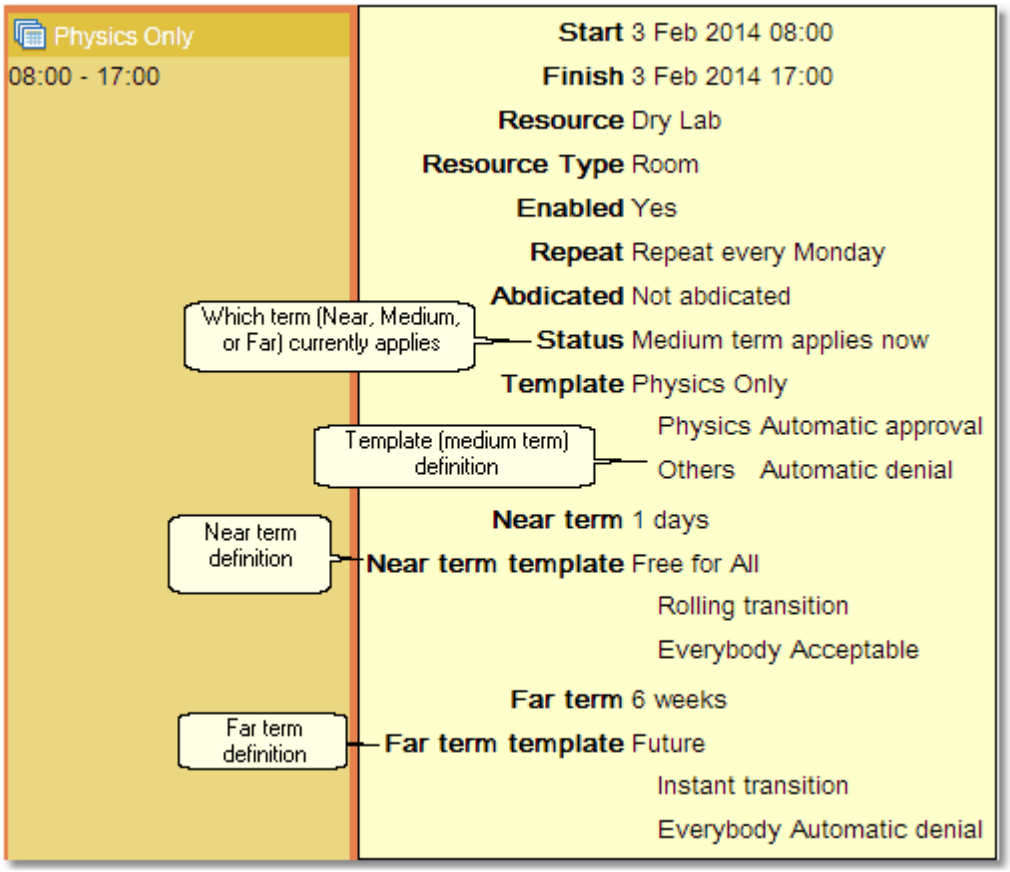

06/05 Mon
07/05 Tue
08/05 Wed
09/05 Thu
10/05 Fri
11/05 Sat
12/05 Sun
dry lab
All bookings set to Acceptable and a warning message pop ups
Biology users Acceptable, Physics users and Everyone Automatically Denied.
Physics users Automatically Approved, Biology and Everyone Automatically Denied
Physics users Acceptable, Biology users and Everyone Automatically Denied.
All bookings set to Acceptable
No Template set, all bookings set to Acceptable
All bookings set to Automaticaly Denied
Prohibited Time
18:00 - 23:55

The *Time Templates Calendar* works in very much the same way as the *Bookings Calendar*. Read the section on Exploring the Bookings Calendar[37] for more details on navigation, *bookmarks*[447], choosing *resources* to display and day/week/month views.

In the above example what is described is the medium term *Time Template* affect for each *Time Template*. As described in How Time Templates Work[215] each *Time Template* may also has have a *near*[448] and *far term Time Templates*[447] configured.

© 2010-2015 Exprodo Software

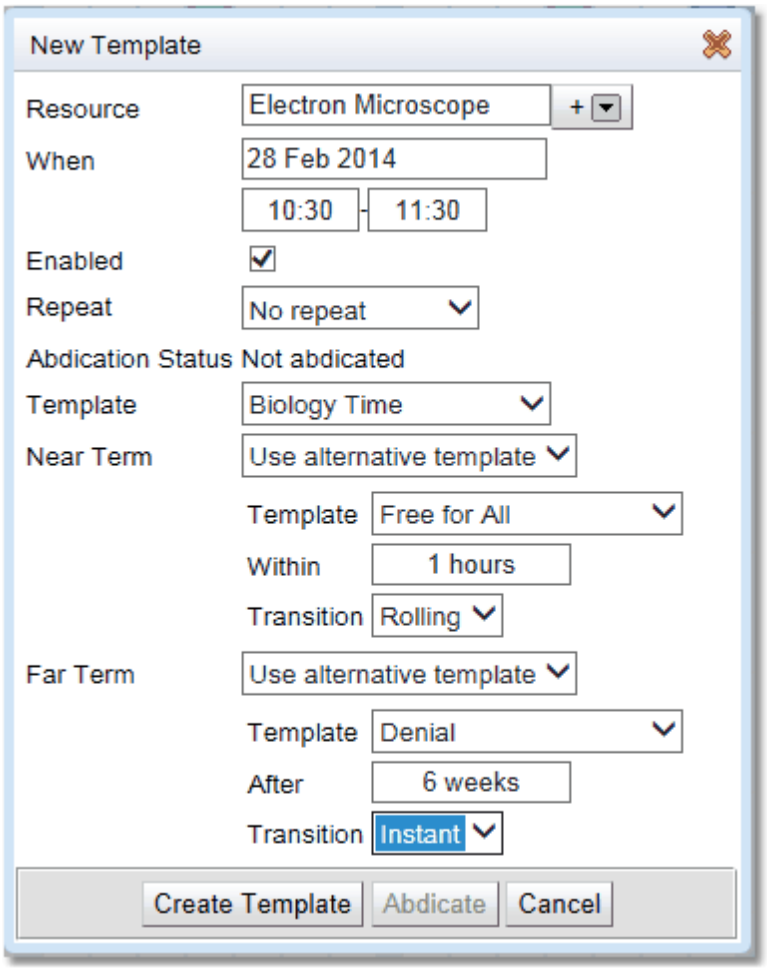
In order to view this information in the *Time Templates Calendar* move the mouse over a *Time Template* and a pop up will appear which will show the complete definition of the *Time Template*.



© 2010-2015 Exprodo Software

Putting Time Templates On The Calendar

To put a *Time Template* on the *calendar*, in the *Time Templates Calendar*, navigate to the date that required for the *Time Template* to appear. Then either single-click in the *calendar* at the time when the *Time Template* needs to start, or click-and-drag to select both the start and end times of the *Time Template*. Both methods will cause the *Time Template* pop-up to be displayed:



This pop-up is where a user will create, edit and view the details of a *Time Template*.

Every *Time Template* must have a *resource*, therefore select which *resource* you want to make a *Time Template* for. Use the drop down arrow to choose the *resource*.

Then adjust the date and times of the *Time Template*.

Decide whether the *Time Template* is to be enabled, disabled *Time Templates* are displayed in the *calendar* but in a lighter colour.

Set up the *repeat* structure for the *Time Template*. For more information on setting up r*epe*ats read the section on Repeat Bookings 53.

Select the *Time Template* required for the medium term.

Select the *Time Template* required for the *Near Term*

Define what is meant by *Near Term*

Decide whether the transition to the new *Time Template* will be **Rolling** or **Instant**. For example if the *Time Template* runs from 08:00 to 20:30, with **Instant** as soon as 08:00 is reached the whole time period becomes available under the *Near Term Template*, with **Rolling** the *Near Term Template* becomes available as the time roles through it. Therefore at 10:00, 08:00 to 10:00 is available under *Near Term* but the rest of the *Time Template* is still the Medium Term

© 2010-2015 Exprodo Software

Select the *Time Template* required for the *Far Term*

Define what is meant by *Far Term*

Decide whether the transition to the new *Time Template* will be **Rolling** or **Instant**.

Once all the details have been entered then press **Create Template**, then the new *Time Template* will be sent to the **Calpendo** server.

Time Templates can be copied and pasted, for fast entry of similar *Time Templates*. Just click on a *Time Template* in the *calendar* and select **Copy** from the drop down menu. Then, click somewhere to create a new *Time Template*, if there's a *Time Template* in the clipboard, its content will be used to populate the *Time Template* pop-up. You can then edit it as required.

Editing Time Templates

Read the chapter on Editing Bookings 54 to see how to edit *Time Templates* on the *calendar*.

6.6.3 Configuring Booking Rules

Whenever a *booking* 446 is created or changed, **Calpendo** will consult its *Booking Rules* 446 to see whether to accept or reject the *booking*, or else give the user a warning about it.

6.6.3.1 How Booking Rules Work

Every time somebody creates or changes a *booking* 446, **Calpendo** checks its *Booking Rules* to see whether the change should be allowed. Unlike *Time Templates* 450 which are applied in the user's web browser, *Booking Rules* are applied in the **Calpendo** server. This key difference has the following consequences:

- *Time Templates* can give feedback to the user before they even try to create a *booking*, whereas *Booking Rules* require the user to enter the *booking* details before any feedback can be given.
- *Time Templates* have limited information available to them, whereas *Booking Rules* have full access to everything in the **Calpendo** database, and even external information available on the internet as well if you need it.

When **Calpendo** checks its *Booking Rules*, it first looks to see which ones apply to the *booking* and the change being made. There are different types of Booking Rules 229, but they all share the same way of identifying which *bookings* they *apply to* 446, and this is described in the next section, Choosing Which Bookings A Rule Applies To 226.

Each *Booking Rule* has to decide whether the *booking* should be rejected or whether it is acceptable. If acceptable, then it must also decide whether to issue the user a warning. When a warning or rejection is issued, the *Booking Rule* responds with two extra things:

- a message that should be given to the user, preferably customised so it is specific to the problem it has encountered.
- for *bookings* whose *status* 447 is **Approved**, the *Booking Rule* can indicate whether it might be acceptable if the *booking's status* were **Requested** instead. This allows the

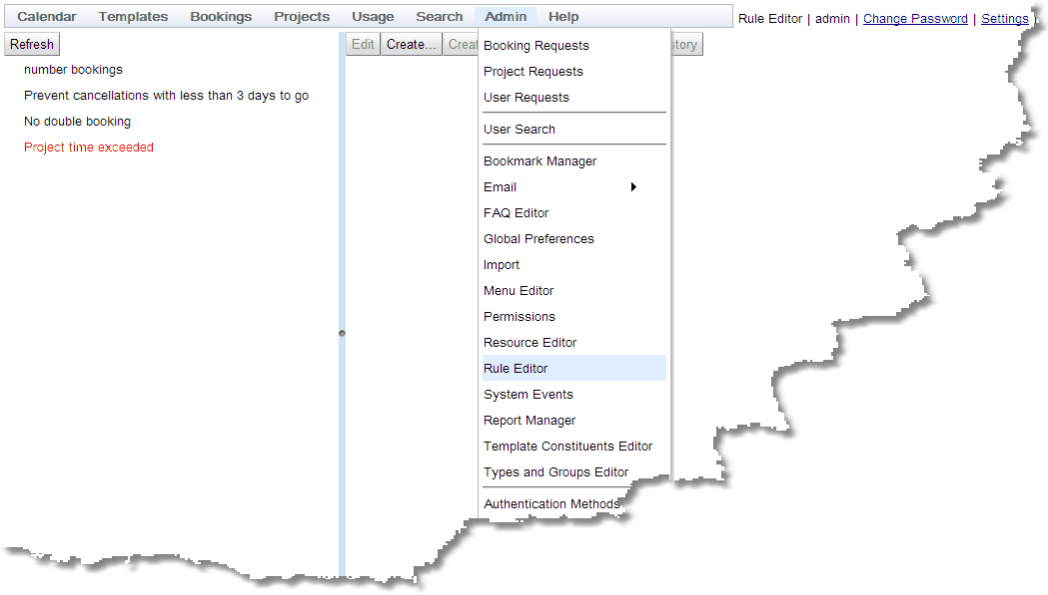
© 2010-2015 Exprodo Software

user the opportunity to downgrade their *booking* to a request, although the option should only be given if the *Booking Rule* would generate a different response to a *booking* request.

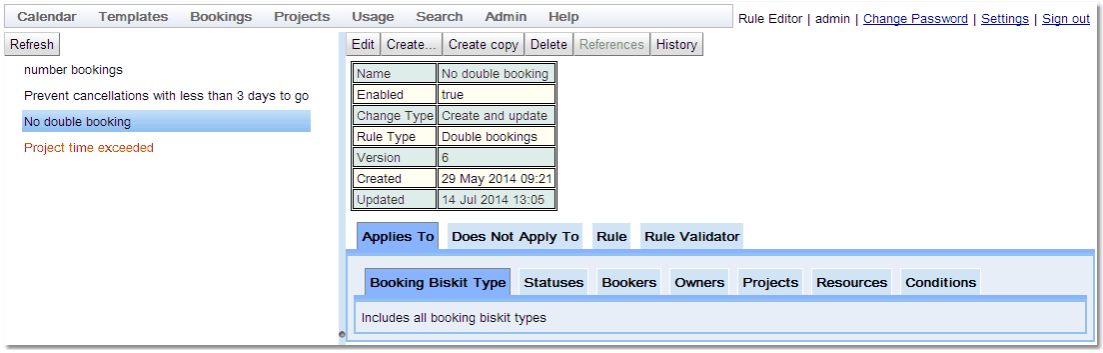
If **Calpendo** finds a *Booking Rule* that rejects the change, then no further *Booking Rules* are run. However, if a *Booking Rule* is found to generate a warning, then further *Booking Rules* are run, just in case there is a *Booking Rule* that would reject the *booking*. Also, **Calpendo** will only show one message to the user, so the order in which the *Booking Rules* run can affect any error message the user is given. The order in which *Booking Rules* are run is controlled by the order they appear in the **Rule Editor.**

6.6.3.2 The Booking Rules Editor

The **Booking Rules Editor** shows all the *Booking Rules* 446 and gives the ability to create, update and delete them. By default, it appears on the menu here:



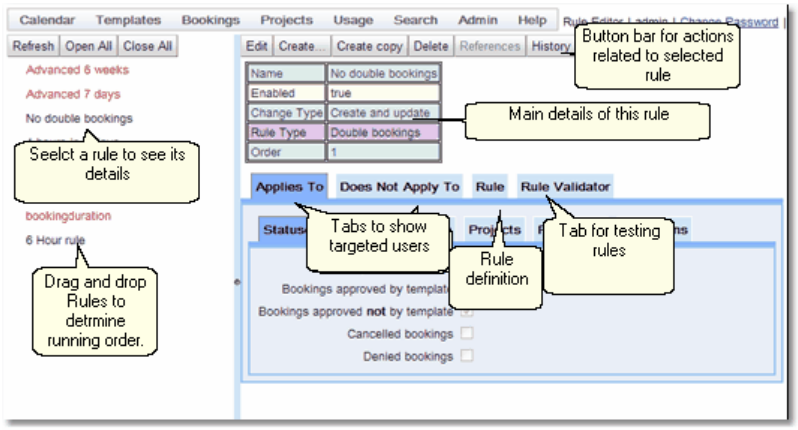
This is what the **Booking Rules Editor** looks like when first opened, disabled *Booking Rules* will be shown in red.



© 2010-2015 Exprodo Software

Booking Rule Details

Click on a *Booking Rule*[446] in the list, to see its details appear on the right:



For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition[96] chapter.

Name	Prevent cancellations with less than 3 days to go
Enabled	true
Change Type	Update
Rule Type	Simple

A *Booking Rule* consists of the following:

Name
The name is only used to provide a way for whoever configures or examines *Booking Rules* to understand the purpose of each *Booking Rule.*

Enabled
The **Enabled** flag is a way for you to create a *Booking Rule* without making it take effect immediately, or to turn one off without deleting it. A disabled *Booking Rule* shows in red in the tree.

Change Type
What type of change to the *booking*[446] will fire this *Booking Rule*. There are three options: **Create**, **Update** and ***Create and Update.***

Booking Rule Type
Which *Booking Rule Type*[446] will be used. See Types of Booking Rule[229] for a full list of the *Booking Rule Types*.

Finally the order the *Booking Rules* will run in is determined by their order in list of *Rules* on the left hand side of the **Rule Editor** starting at the top.To change this order drag and drop a *Rule* to its new position in the list.

© 2010-2015 Exprodo Software

6.6.3.3 Choosing Which Bookings A Rule Applies To

This is split into two sections:

Applies To[446]: Determines the *bookings*[446] that the *Booking Rule*[446] applies to.

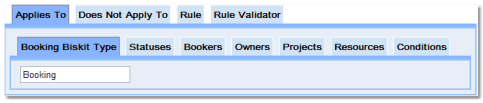
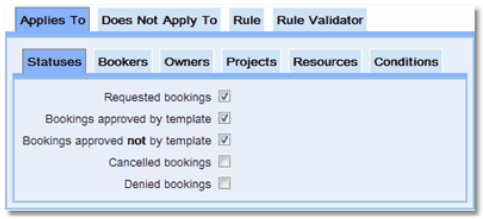
Does Not Apply To[447]: Determines the *bookings* that the *Booking Rule* does not apply to.

By default a *Booking Rule* will apply to all *bookings*, so if there is nothing in both *Applies To* and *Does Not Apply To* then the *Booking Rule* will apply to all *bookings*. Therefore using *Does Not Apply To* only the *bookings* the *Booking Rule* does not apply to can be specified. Also combine the two types and a *Booking Rule* will apply to a *booking* as long as its accepted by the *Applies To* but not accepted by the *Does Not Apply To*. Of course *Applies To* can be used by itself to figure out which *bookings* a *Booking Rule* will apply to.

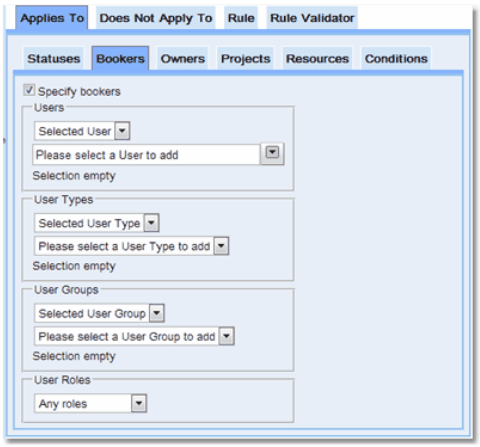
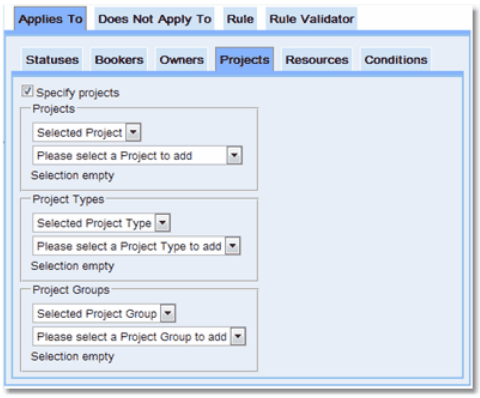
Applies To

When running *Booking Rules*, **Calpendo** must decide which *Booking Rules* apply for a given *booking* change that a user is trying to make. To do this, **Calpendo** first looks to see which *Booking Rules* are marked as **enabled**. Then, it looks at whether the *Booking Rule* says it applies when a *booking* is created, when a *booking* is updated, or both.

Beyond those simple *properties*[448], every type of *Booking Rule* provides additional tabs in which you can identify the *bookings* it should apply to as follows:

Tab	Description	
Booking Biskit Type	An administrator can define a number of **Booking** *Biskit Types*[446] to allow for different *booking* information to be stored for *resources*[449]. This tab allows the administrator to decide which of the **Booking** *Biskit Types* the *Booking Rule* will work with. A *Booking Rule* will also work on any **Booking** *Bisket Type* that inherits from the selected one.	
Statuses	Every *booking* has a *status*[447] which is one of **Requested**, **Approved**, **Cancelled** or **Denied**. For those *bookings* whose *status* is **Approved**, **Calpendo** also remembers whether the booking was approved by a *Time Template*[450], or approved manually. The **Statuses** tab allows a choice of which *booking statuses* this *Booking Rule* should apply to, and if *approved bookings*[446] are required, whether to apply to *bookings* approved by *Time Template*, manually or both.	

© 2010-2015 Exprodo Software

Bookers	The **Bookers** tab specifies for which registered *booker*[446] of these *bookings* you want this *Booking Rule* to apply to. First, tick the **Specify Bookers** box in order to activate this tab. Once activated specify the *booker* by identifying particular users, *user types*[450], *user groups*[450] or *user roles*[450]. If the *bookers* to be accepted are specified using a mixture of specified users, *user types*, *user groups* or *user roles*, then as long as the *booking* being changed has a *booker* that passes any of the *booker* specifications, then the *booker* will be accepted.	
Owners	This works exactly the same as booker, but specifies who owns the bookings the *Booking Rule* is to apply to. This works exactly the same as **Bookers**, but specifies who owns the *bookings* the *Booking Rule* is to apply to.	
Projects	Very much like **Bookers** and **Owners**, this tab specifies the *projects*[448] a *booking* must have been made for in order for this *Booking Rule* to apply, by giving specific *projects*, *project types*[448] and/or *project groups*[448].	
Resources	This works in a similar fashion to *projects* but it specifies the *resources*[449] the *booking* will apply to by giving specific *resources*, *resource types*[449] and/or *resource groups*[449].	

© 2010-2015 Exprodo Software

Tab	Description	
Conditions	The **Conditions** tab works just like the Conditions [98] used elsewhere. This allows custom *conditions* [447] to be provided about the state of the *bookings* you want the *Booking Rule* to apply to. Note that all the other tabs implicitly provide specifications of what the *booking* would look like after the change being made, if it were allowed. For example, if somebody changes the *project* on a *booking*, then *Booking Rules* that apply only to certain *projects* will apply as long as it matches the new *project*, not the old one. However, *conditions* allow explicit reference to the state of the *booking* both before and after the change.	

Within each activated tab (**Statuses**, **Bookers**, **Owners**, **Projects**, **Resources**) if a *booking* passes any one of the sections set up in that tab then the *booking* will be accepted for that tab. So if there are items in more than one section of a individual tab (i.e. groups, types etc.) then if any of the sections within a tab is true, then the tab is true. This if a *Booking Rule* is applied to *project type* P and *project group* G, then the *Booking Rule* applies to *projects* of Type P or any *projects* in *project group* G. (The *project* doesn't have to be both of *Type* P and in *Group* G for it to pass, just one of them). If may be required that the *Booking Rule* may only want to apply when the *project* is in a particular *group* and also has a particular *project type*. In order to achieve this, specify the *project group* in the **Projects** tab, and then add a *Condition* in the **Conditions** tab to require that the *project type* be what is required.

In order for a *booking* to be finally accepted it must be accepted in all the activated tabs. So if **Bookers** and **Resources** are activated and there is also a **Condition** then the *booking* must be accepted in each of the three tabs in order to be finally accepted. If it is accepted by only one or two of these tabs then it will be rejected.

© 2010-2015 Exprodo Software

Does Not Apply To

Finally, as well as allowing the specification of *bookings* the *Booking Rule* does *apply to*, it also allows the same sort of specification for those it *does not apply to*. Sometimes it may be easier to say which *bookings* a *Booking Rule* shouldn't *apply to*, and sometimes it's easier to target the *bookings* required by specifying some things in the **Applies To** section, and some other things in the **Does Not Apply To** section.

Does Not Apply To works in exactly the same way as *Applies To.*

For example, a *Booking Rule* may be needed that is to be applied to most people, but a small number of people should be exempt (perhaps because they are administrators or heads of group). Implement this by creating a *user group* with all the people such a *Booking Rule* should apply to. However, *user groups* tend to work best when a small number of users belong to them - otherwise maintaining the *groups* with the right membership can become onerous. It's therefore easier to maintain a *user group* of the people that should be exempt, and adding them to the **Does Not Apply To** section.

If users are classified by assigning them all a *user type*, then create a *Booking Rule* which targets a particular *user type* in the **Applies To** section, but then removes some users from that selection by the use of a **Does Not Apply To** specification of users.

To see examples of what can be done with this way of choosing which *bookings* a Booking Rule should apply to, please see the examples shown for Simple Booking Rule 230.

One problem to be wary of is setting **Admin** in the **Bookers** of **Does Not Apply To** and having something in the **Condition** tab as well and expecting **Admin** too always be exempt. They will only be exempt if the *Condition* also applies. In this case add a *Condition* of **New Value of booker.roles includes any Admin**.

6.6.3.4 Types of Booking Rule

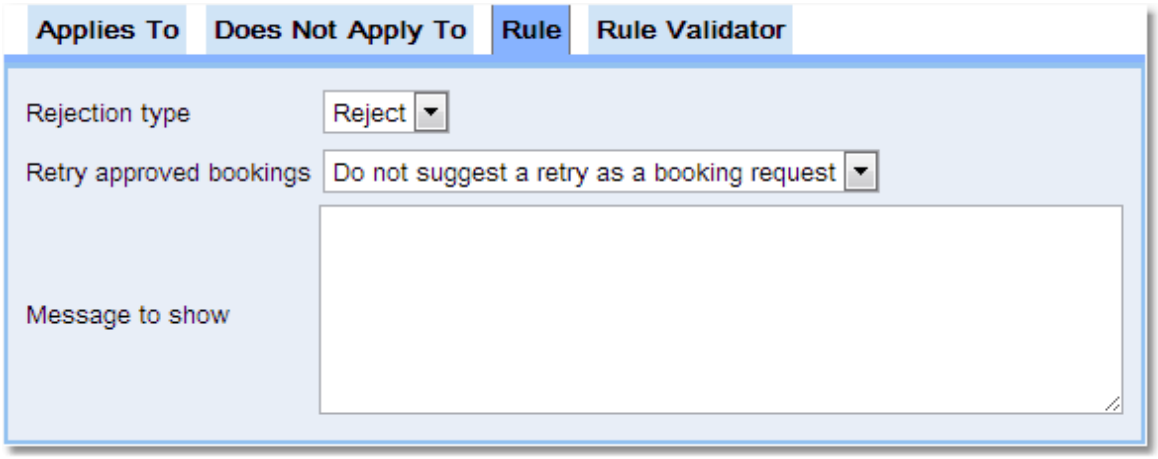
As previously mentioned, all **Calpendo** *Booking Rules* 446 specify which *booking* 446 changes they apply to. Beyond that, the different types of *Booking Rule* vary in their complexity. At one end of the scale is the Simple Booking Rule 230, which always rejects or warns about a *booking*, with its real logic being restricted to the selection of the *bookings* it applies to. At the other end of the scale is the Advanced Booking Rule 256 which requires a programmer to configure, but allows full access to the **Calpendo** database and beyond.

In between the two extremes, there are *Booking Rules* that specialise in particular areas. It is expected that more specialist *Booking Rule Types* 446 will be developed over time. In many cases, these new *Booking Rule Types* are likely to start life as an **Advanced Booking Rule**. The following sections describe each of the different types of *Booking Rule*.

© 2010-2015 Exprodo Software

6.6.3.4.1 Simple Booking Rule

Beyond the standard settings for selecting which *bookings*[446] it applies to, a **Simple Booking Rule** only specifies three things:



Property	Description
Rejection Type	This sets up whether to reject the *booking* or just post a warning.
Retry approved bookings	An indication to be sent to the user of whether to retry sending the *booking* as a **Request** (only applicable when the *booking* received is **Approved**).
Message to show	The message to be displayed to the user if the *booking* is rejected or a warning is issued.

Please note that there is no intelligence here to decide whether or not to reject *booking* or give the user a warning - all the logic supported by a **Simple Booking Rule** is in the selection of the *bookings* the *Booking Rule*[446] applies to. While it may seem that this would make a **Simple Booking Rule** somewhat limited in scope, you can achieve quite a lot with it, as shown by the following examples.

© 2010-2015 Exprodo Software

Example: Disallowing Bookings More Than 6 Weeks In Advance

Suppose that the *resources*[449] are very busy, and people are responding by *booking* well into the future, just in case they will have a need. So its decided to stop this behaviour by disallowing *bookings* more than 6 weeks away.

This result can be achieved with Permissions[278], although that wouldn't allow a nice error message to the user - they would just receive the standard **Permission Denied** message.

This can also be done with a Time Template[215] as well as a *Booking Rule*, and it's probably better to do it with a *Time Template*[450] so that the user would get feedback before they try to make such a *booking*. However this example may be helpful.

To do it, create a **Simple Booking Rule** with **Change Type** of **Create and Update** and add this as the *Condition*[447]:

Section	Tab	Description
Applies To	**Conditions**	**New Value** of **dateRange.start** later than now plus 42 days to the minute

This adds 6 weeks to the current time (the time when the user tries to make a *booking*), and then tests whether the *booking* starts (**dateRange.start**) after that time, when both times are rounded to the nearest minute. If the *booking* starts after the 6 week time, then the *condition* is **true**, and so the *Booking Rule* applies.

Please note that if *repeats*[449] are allowed the **Global Preference->Bookings->Number of Days of Bookings** to check for *repeat bookings* must be larger than the number of days you are checking for by 1-2 weeks to stop *repeat bookings* going after that time, depending on what type of *repeat* is being used. A better option would be:

Section	Tab	Description
Applies To	**Conditions**	**New Value** of **repeat.finish** later than now plus 42 days to the day

© 2010-2015 Exprodo Software

Example: Physicist Projects Can Book 4 Weeks In Advance, Biologists 6 Weeks

This expands on the previous example, but this time we want to have a different time limit depending on the project used to make the *booking*. Again, this could be done with *Time Templates*, but we'll show how to do it with *Booking Rules*.

Create two **Simple Booking Rules** this time with **Change Type** of **Create and Update**, one to prevent physicists from *booking* too far ahead, and one to prevent biologists from *booking* too far ahead. The physicists' *Booking Rule* would look like this:

Section	Tab	Description
Applies To	**Conditions**	**New Value** of **dateRange**.start later than now plus 28 days to the minute
Applies To	**Projects**	Add the **Physicist** *project type* 448 to the list of targeted *projects* 448

and the biologists' *Booking Rule* would look like this:

Section	Tab	Description
Applies To	**Conditions**	**New Value** of **dateRange**.start later than now plus 42 days to the minute
Applies To	**Projects**	Add the **Biologist** *project type* to the list of targeted *projects*

This assumes that *project types* are used to classify the *projects*. An alternative would be to name the *projects* individually in the *Booking Rule*, or else create *project groups* 449 and use those instead.

© 2010-2015 Exprodo Software

Example: Special Projects Can Book Any Time In Advance

Now suppose that there is a *project group* that lists **Special Projects** that should be allowed to book as far into the future as it wants to. A good example of this is if your *resource* 449 needs regular maintenance and so you set up a **Maintenance** *project*. Most probably associate a small number of users with this *project*, to restrict who can create such *bookings*.

Modify the physicist and biologist *Booking Rules* from the last example so that each of them adds to the **Does Not Apply To** section:

Section	Tab	Description
Does Not Apply To	**Projects**	Add the **Special Projects** *project group* to the list of targeted *projects*

The effect of this is that whenever a *booking* is made for a *project* in the **Special Projects** *project group*, the *Booking Rules* we created above no longer apply, and so it removes the restriction about how far into the future such *bookings* can be for.

© 2010-2015 Exprodo Software

Example: Special Projects And Admins Can Book Any Time In Advance

Now suppose that as well as allowing **Special Projects** to be allowed to book as far into the future as they want to, administrators need to be able to make *bookings* for any *project* well into the future. The users to be excluded from the time limit could be specified by listing them individually, by their *user role*[450] or by creating a *user group*[450] for it. Any of these would be sensible, with the choice depending on how **Calpendo** should be set up to meet your general requirements.

Let's suppose we want to allow users with the **Admin** *user role* to book well into the future for any *project*. Then the physicist and biologist *Booking Rules* could be changed from the previous example so that their **Does Not Apply To** sections contain this:

Section	Tab	Description
Does Not Apply To	**Projects**	Add the **Special Projects** *project group* to the list of targeted *projects*
Does Not Apply To	**Bookers**	Add **Admin** to the **bookers** *user roles* selection.

However, this is probably not what is required. With this prescription, any booking whose **Booker** *property*[448] is set to **Admin** and whose *project* is in the **Special Projects** group will be allowed to book well into the future. For a *Booking Rule* such as this, it may be required to exclude a *booking* from the *Booking Rule* if the *project* is "special" or if the *booking* were made by an administrator. To achieve this, replace the **Bookers** entry with a *condition*, like this:

Section	Tab	Description
Does Not Apply To	**Projects**	Add the **Special Projects** *project group* to the list of targeted *projects*
Applies To	**Conditions**	***New Value*** of ***booker.role*** does not contain any **Admin**

Note that we also put the *condition* on the **Applies To** section. Had we used a *condition* in the **Does Not Apply To** section, then we would still have required all of the *Does Not Apply To*[447] to be applicable for the exclusion to work.

There's also another caveat to this example: if the Permissions[278] do not stop non-administrators from making a *booking* with the **booker** *property* set **Admin**, then a non-administrator could work around this *Booking Rule* by setting the **booker** to somebody else when they create the *booking*. This practice can be stopped with *Permissions*[448], but can also be stopped by making the *condition* in the *Booking Rule* checked the **user** that performed the change, rather than the **booker** *property* of the *booking*. This is shown in the next example.

© 2010-2015 Exprodo Software

Example: Prevent Cancellations With Less Than 24 Hours To Go

Depending on the policies employed at the facility, last-minute cancellations may need to be prevented apart from those made by an administrator. This will require a *Booking Rule* that targets changes to *bookings* that result in the *booking status*[447] changing to **Cancelled** from something else, that was made within 24 hours of the *booking* time, and that the change was not made by an administrator.

This can be done with a **Simple Booking Rule** with **Change Type** of **Update** that has the following added to the **Applies To** section:

Section	Tab	Description
Applies To	**Statuses**	Tick only **Cancelled** *Bookings*
Applies To	**Resources**	Choose the *resources* the *Booking Rule* should apply to
Applies To	**Conditions**	**Old Value** of **dateRange**.**start** earlier than Now Plus 24 hours to the minute
Applies To	**Conditions**	**Old Value** of **status** not equal to **Cancelled**
Applies To	**Conditions**	***New Value*** of **status** equals **Cancelled**
Applies To	**Conditions**	*Meta-property*[448] **user.roles** does not include any **Admin**

There are some important things to note about these *conditions*:

- By checking the *meta property* called **user, person** that makes the change is being checked, regardless of who is set to be the *booker*[446] or *owner*[448] of the *booking*.
- We could have moved the *user.roles* check to the **Does Not Apply To** section by negating the sense of the *condition*. That is, by adding a *condition* to the **Does Not Apply To** section that said *Meta property* **user.roles** includes ***A*dmin**. This would have been functionally identical, as long as there were no other entries in the **Does Not Apply To** section.

In the **Rule Tab**, enter a message to be displayed to people who try to cancel within 24 hours of the *booking*.

© 2010-2015 Exprodo Software

6.6.3.4.2 Double Booking Rule

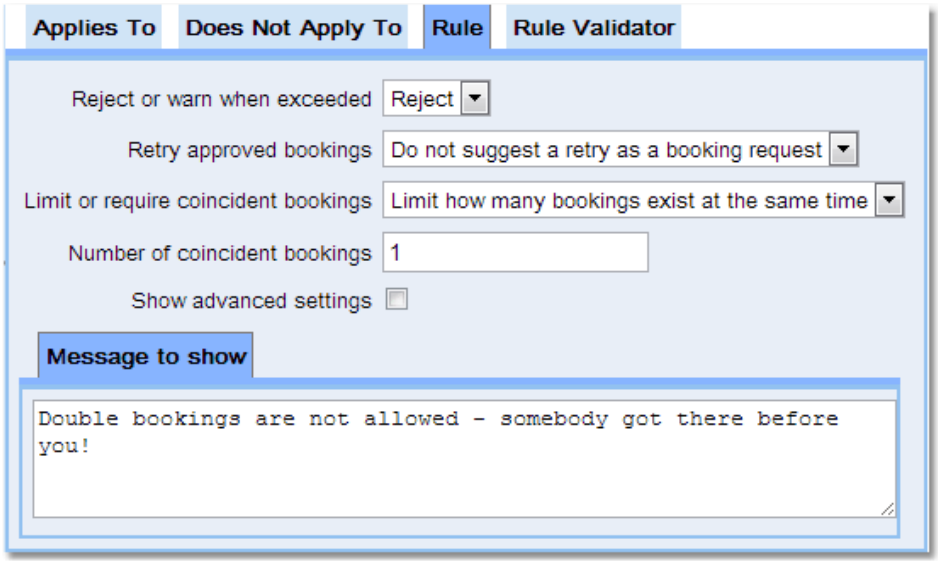
Prevention of double *bookings* [446] is clearly something that a *booking* system ought to be able to do. **Calpendo** doesn't automatically prevent double *bookings*, but instead provides a **Double Booking Rule** to allow the system to define precisely how it ought to behave in this area. For example, to create a *Booking Rule* [446] that doesn't allow more than one *booking* for the same *resource* [449] at the same time. That might seem sensible, but it's not always that simple.

Suppose there is a computer room that people can book, and there are 10 computers in there. **Calpendo** could be configured to handle this by creating 10 separate *resources*. However, the alternative is to configure **Calpendo** with a single *resource*, to represent the computer lab itself, and then create a **Double Booking Rule** that allows up to 10 *bookings* for it at the same time.

Another example that shows that double *bookings* are not always as simple as you might always think: suppose there is a MRI scanner with an adjoining room that can be used to treat patients. Users are allowed to book the MRI scanner whenever they like for subjects that are healthy, and therefore not expected to require the use of the adjoining treatment room. However, if there is a need to scan somebody that requires treatment facilities to be available because there's a risk they may require such help, then the user can't book the MRI scanner when somebody already has a *booking* for the treatment room.

This is also a type of double *booking* because it relates to *bookings* that exist at the same time, albeit across different *resources*. This can be implemented in a couple of ways in **Calpendo**. When booking for an MRI scanner for a patient that may require the treatment room **Calpendo** can:

- only allow the MRI scanner to be booked if there isn't already a *booking* for the treatment room, and also do not allow the treatment room to be booked if there's an MRI *booking* for a patient that may require the treatment room.
- only allow the MRI scanner to be booked if the same *project* [448] or *booker* [446] already has a *booking* for the treatment room.


Applies To
Does Not Apply To
Rule
Rule Validator
Reject or warn when exceeded
Reject
Retry approved bookings
Do not suggest a retry as a booking request
Limit or require coincident bookings
Limit how many bookings exist at the same time
Number of coincident bookings
1
Show advanced settings
Message to show
Double bookings are not allowed - somebody got there before you!

© 2010-2015 Exprodo Software

The default settings when creating a **Double Booked Rule** are configured for the simple case - to stop multiple *bookings* for the same *resource* at the same time.

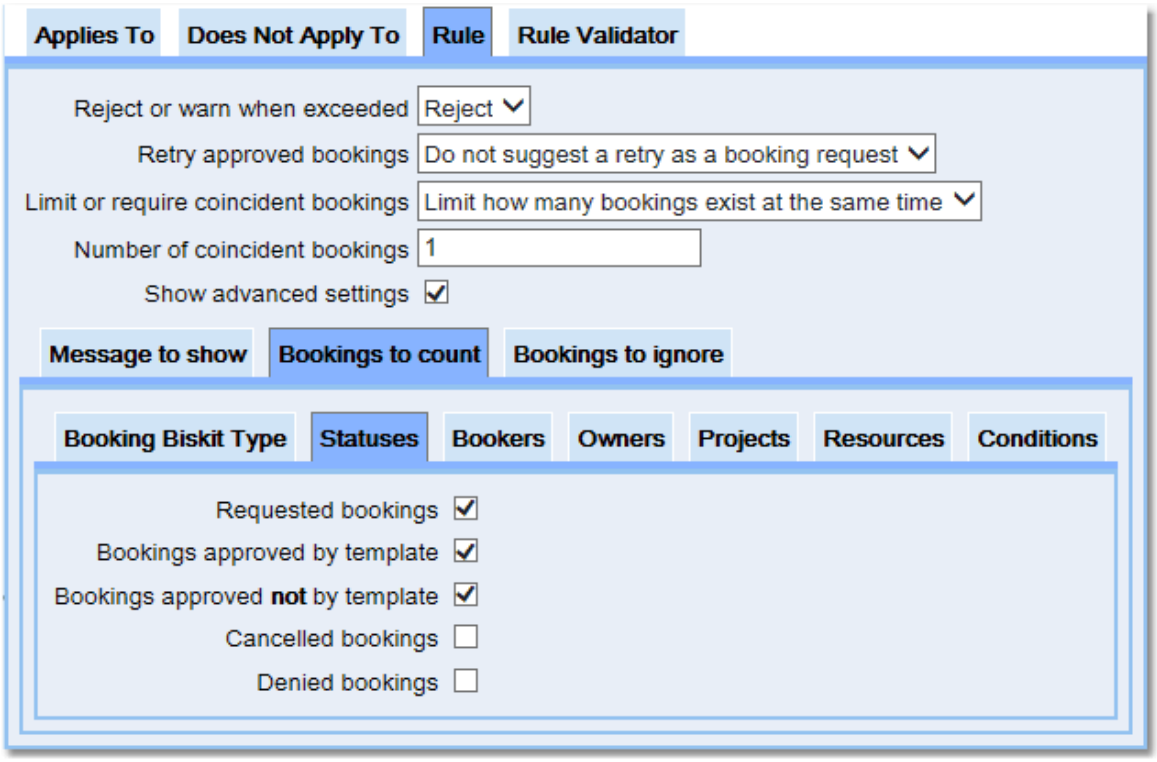
Calpendo allows the handling of more advanced methods as shown in this table:

Property	Description
Rejection Type	Decides whether to reject the *booking* or just post a warning.
Retry approved bookings	An indication to be sent to the user of whether to retry sending the *booking* as a **Request** (only applicable when the *booking* received is **Approved**).
Limit or Require coincident bookings	Choose whether the *Booking Rule* should operate in a mode that limits the number of *bookings* that can coexist (the standard case), or whether the *Booking Rule* should require at least a given number of *bookings* to exist beforehand (for example to ensure that the treatment room is booked before being allowed to book the MRI scanner)
Number of coincident bookings	The standard **Double Booking Rule** sets the number of *bookings* allowed to one. This means that there can only be one *booking* at a time. In the case of the computer lab with ten computers, this could be set to ten.
Message to show	The message to be displayed to the user if the *booking* is rejected or a warning is issued.
Bookings to count	This tab is only shown when the **advanced settings** option is selected. This is described in detail below.
Bookings to ignore	This tab is only shown when the **advanced settings** option is selected. This is described in detail below.

© 2010-2015 Exprodo Software

Advanced Settings

Turning on the **advanced settings** option in the **Double Booking Rule** displays two extra tabs: **Bookings To Count** and **Bookings To Ignore**. These tabs are not required in the simple case, where we only want to prevent multiple *bookings* for the same *resource* to coexist. They *are* required when *bookings* for one *resource* need to influence *bookings* for another *resource*.



The **Bookings to Count** and **Bookings to Ignore** work in exactly the same way as *Applies To*[446] and *Does Not Apply To*[447], read the chapter on Choosing Which Bookings A Rule Applies To[226] for more information on how the sub tabs work.

In the MRI scanner and treatment room example, there are two ways to implement it: preventing a *booking* for a patient that may require the treatment room when the treatment room is already booked or requiring that the treatment room already be booked by the same *project*. In each case, we need to identify which MRI *bookings* might require the treatment room. For that, use the Types And Groups Editor[266] to create a *booking type*[447] to represent the type of subject being scanned. Classify subjects according to whether they are healthy volunteers, walking patients, wheelchair patients, or bedridden patients. Now suppose that healthy volunteers do not require the treatment room to be available, but the other types do.

© 2010-2015 Exprodo Software

Here's the first option for how to set this up in **Calpendo**, by preventing the treatment room from being booked at the same time as the MRI scanner:

Property	Setting
Applies To	Add *condition*[447]: *type* not equal to **Healthy Volunteer**
Applies To	On the **Resources** tab, specify the **MRI Scanner**
Reject or warn when exceeded	Reject
Retry approved bookings	Do not suggest retry as a *booking* request
Exclusivity	Limit how many *bookings* can exist at the same time
How many bookings allowed	One
Show advanced settings	Yes
Bookings To Count	On the **Resources** tab, change **Matched Resource** to **Treatment Room**
Bookings To Ignore	Not required

This means that any *booking* for the MRI scanner, where the *booking type* is not set to **Healthy Volunteer** will trigger the *Booking Rule*. The *Booking Rule* will then count how many *bookings* there are for the treatment room at the same time as the prospective MRI scanner *booking*. The limit for the number of allowed *bookings* is one, and this always includes the prospective *booking* (in this case the one for the MRI scanner). The net result is that MRI scanner *bookings* will be rejected when being made for non-healthy volunteers if there's a *booking* for the treatment room at the same time.

If it is implemented in this way, then a similar *Booking Rule* would be needed that prevented *bookings* for the treatment room when there are coincident *bookings* for the MRI scanner for non-healthy subjects.

© 2010-2015 Exprodo Software

The second option for how to set this up in **Calpendo** is to require a pre-existing *booking* for the same *project* in the treatment room. Here's what this looks like:

Property	Setting
Applies To	Add *condition*: *type* not equal to **Healthy Volunteer**
Applies To	On the **Resources** tab, specify the **MRI Scanner**
Reject or warn when exceeded	Reject
Retry approved bookings	Do not suggest retry as a *booking* request
Exclusivity	Require a number of pre-existing *bookings*
How many bookings allowed	One
Show advanced settings	Yes
Bookings To Count	• On the **Resources** tab, change *Matched* **Resource** to **Treatment Room** • On the **Projects** tab, tick **Specify projects** and change **Projects** from **Any Project** to **Matched Project**
Bookings To Ignore	Not required

The **Matched Project** setting means that, when **Calpendo** looks for pre-existing *bookings*, it will only include *bookings* that have the same *project* as the prospective one being created (in this case the *project* on the MRI scanner *booking*). This means that, if there's a *booking* for the same *project* in the treatment room, we're going to assume that the treatment room has been pre-booked for the purpose of providing back-up in case the patient requires it.

Bookings To Ignore

The previous examples showed how to use **Bookings To Count** to specify which *bookings* **Calpendo** should count when looking for *bookings* at the same time as the one being created or modified. The **Bookings To Ignore** tab provides an extra option to the way the counting is done, by allowing the specification of *bookings* that are not going to be counted. This is analogous to the way one can specify *Applies To* and *Does Not Apply To* settings.

© 2010-2015 Exprodo Software

Handling Holidays

Some *resources* require the presence of people while they are being used. For example, an MRI scanner requires radiographers. Sometimes, suitably qualified people book the scanner for their own use and don't require an additional radiographer, but on other occasions, *bookings* require additional support. When radiographers are on holiday, the *booking* system should be able to provide suitable information to anybody making a *booking* throughout the holiday period. There are several ways to do this with **Calpendo**, each with their own benefits and drawbacks.

Create a *Time Template* 450 that issues warnings to everyone, and then add entries to the *Time Templates Calendar* 450 whenever there's a holiday. Or, create a Holiday Booking Rule 244. Finally, do it with a **Double Booking Rule** as follows:

1. Create a new *resource* where holiday information is stored. Let's call it **Radiographer Leave** for this example.
2. Create a **Double Booking Rule** that applies to MRI scanner *bookings*, and has its **Bookings To Count** set to count up the number of *bookings* for **Radiographer Leave**.
3. Set the exclusivity to limit how many bookings can exist at the same time and the **How many bookings allowed** to one.
4. Set the **Reject or warn when exceeded** to **warn**
5. Enter a message that there is limited radiographer cover (to show to the user).

Now, when a radiographer plans a holiday, they create a *booking* for the **Radiographer Leave** *resource*. Whenever somebody then makes a *booking* for the *MRI* scanner at the same time, the **Double Booking Rule** will give a warning to the user. The nice thing about implementing holiday handling with this method is that it's very easy to delegate the responsibility for giving the warnings to the radiographers, and there's an entry in the *calendar* 446 so everyone can see when there is no radiographer cover available.

© 2010-2015 Exprodo Software

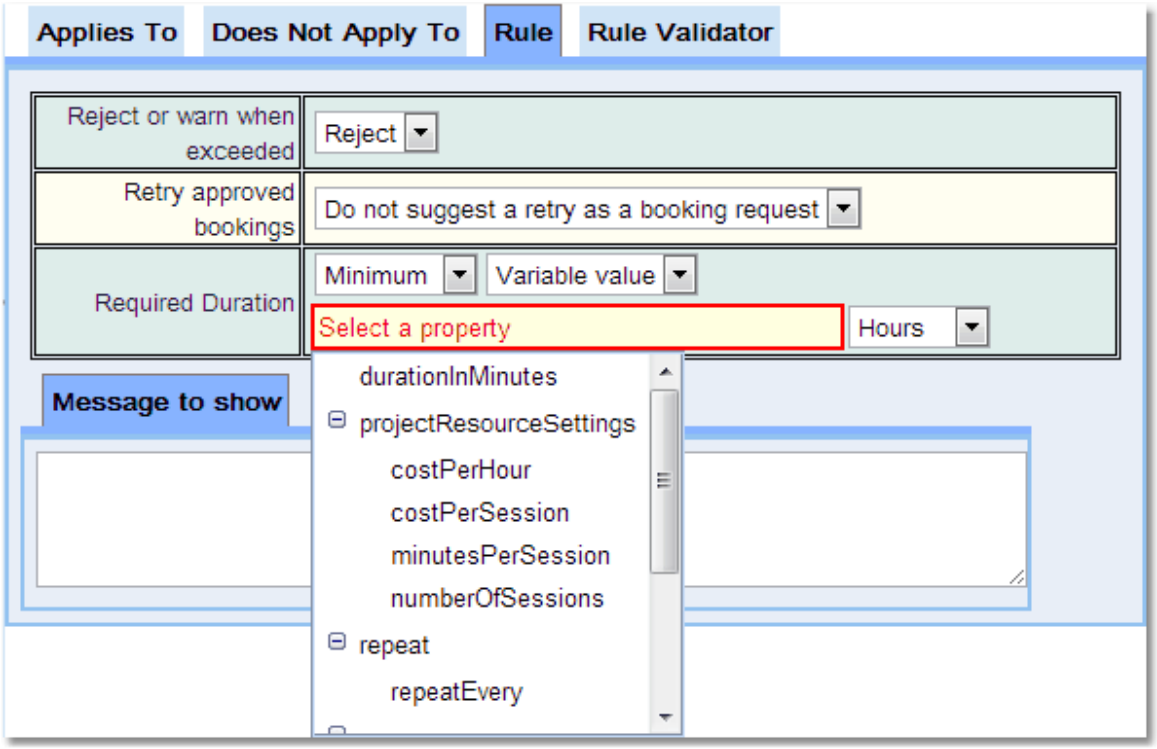
This table compares the methods for displaying warnings to users during a holiday period:

Method	Pros	Cons
Time Template	Provides visual feedback for the duration of the holiday in the background of the *Bookings Calendar.*	• No way to see who is on holiday or how many people are on holiday. • Either need radiographers to modify *Time Templates* 449, or an **Admin** has to set up holidays on behalf of radiographers.
Holiday Booking Rule	Conceptually easy.	• No visual feedback before making the *booking*. • New *Booking Rule* needed for each holiday, or else modify the old *Booking Rule* to add more holidays. • Either need radiographers to modify *Booking Rules*, or an **Admin** has to set up holidays on behalf of radiographers.
Double Booking Rule	• Radiographers are given their own "leave calendar". • They can see and edit their own leave information. • Other users can see radiographer leave information.	Conceptually a little more difficult to set up than the other approaches, although once configured, it's easy for radiographers to add holidays and for users to comprehend.

© 2010-2015 Exprodo Software

6.6.3.4.3 Booking Duration Rule

The **Booking Duration Rule** allows limits to be set on how long *bookings*[446] can be for certain *resources*. Set a minimum or maximum time for *bookings* or specify exactly how long they may be.



Property	Description
Reject or warn when exceeded	Decide whether to reject the *booking* or just post a warning.
Retry approved bookings	An indication to be sent to the user of whether to retry sending the *booking* as a **Request** (only applicable when the *booking* received is **Approved**).
Required Duration	Set the duration to be a minimum, maximum or exactly a specific time period. The time period can be a **Fixed Value** that is specified or a **Variable Value**, something that is read from a *property*[448] accessed through the *booking*.
Message to show	The message to be displayed to the user if the *booking* is rejected or a warning is issued.

© 2010-2015 Exprodo Software

6.6.3.4.4 Holiday Booking Rule

A **Holiday Booking Rule** is very similar to a **Simple Booking Rule**, but also allows a list of dates to be specified. The *Booking Rule*[446] will reject or warn for any *booking*[446] that is on one of the dates provided. This is intended to be used so that there is a *Booking Rule* that's easy to set up that will either reject or produce a warning on particular days, for holidays, equipment servicing or anything else that might be needed.

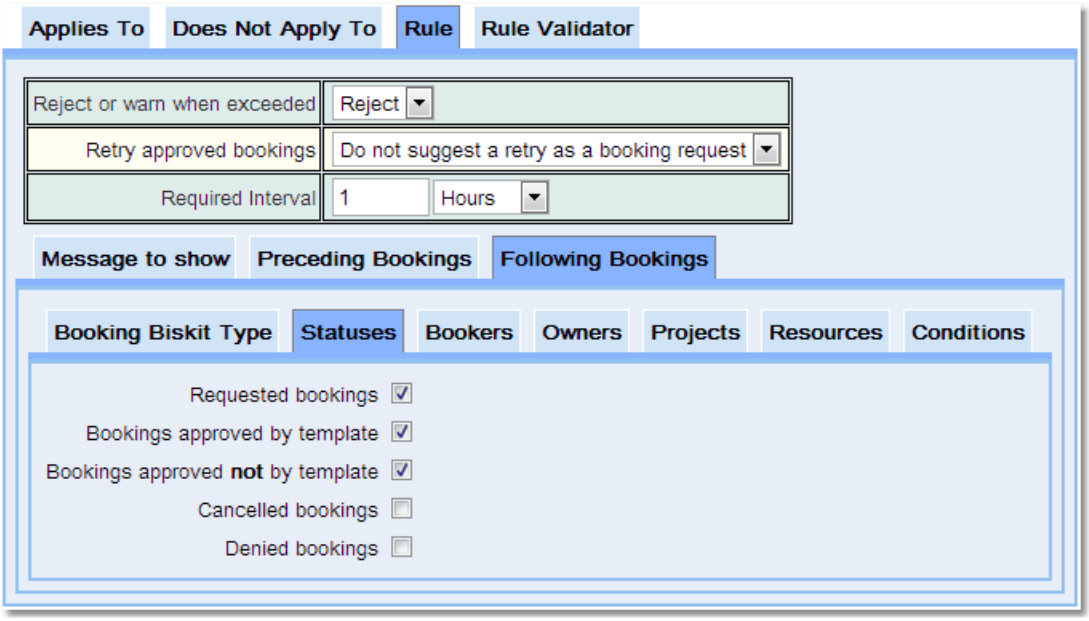
Note that there is similar functionality using either a **Double Booking Rule** or *Time Templates*[450]. Which implementation is preferred will depend on the circumstances. See Handling Holidays[242] for a comparison of the available methods.

Property	Description
Rejection Type	Decide whether to reject the *booking* or just post a warning.
Retry approved bookings	An indication to be sent to the user of whether to retry sending the *booking* as a **Request** (only applicable when the *booking* received is **Approved**).
Message to show	The message to be displayed to the user if the *booking* is rejected or a warning is issued.
Select holiday date	Allows selection of the holiday dates during which this *Booking Rule* will either reject *bookings* or give a warning.

© 2010-2015 Exprodo Software

6.6.3.4.5 Booking Interval Rule

The **Booking Interval Rule** lets the administrator specify that the time between some (or all) *bookings* must be at least a particular time. This can be used to enforce gaps between all *bookings*, or to prevent *projects*[448] that overheat the scanner from following each other too closely.



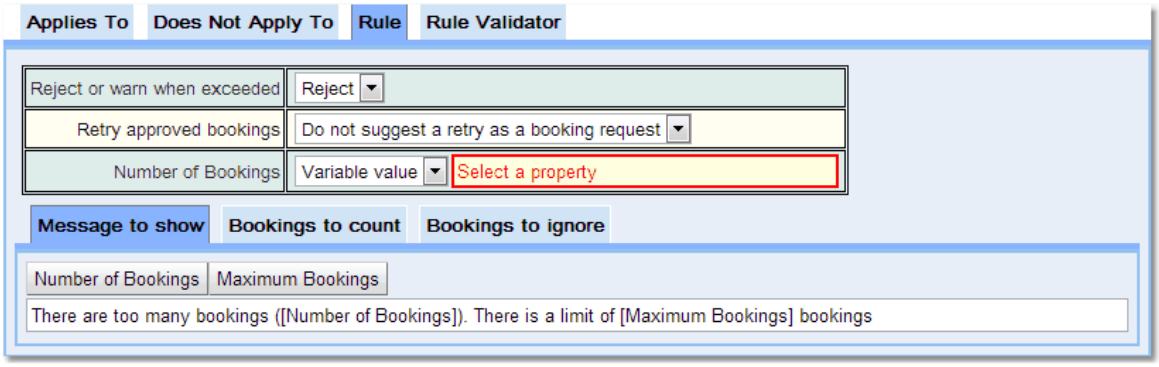
Property	Description
Reject or warn when exceeded	Decide whether to reject the *booking* or just post a warning.
Retry approved bookings	An indication to be sent to the user of whether to retry sending the *booking* as a **Request** (only applicable when the *booking* received is **Approved**).
Required Interval	Time between *bookings*
Message to show	The message to be displayed to the user if the *booking* is rejected or a warning is issued.
Preceding Bookings	The **Preceding Bookings** and **Following Bookings** work in exactly the same way as *Applies To*[446], read the chapter on Choosing Which Bookings A Rule Applies To[226] for more information on how the sub tabs work. The only difference is, it is setting up *bookings* to proceed the *booking* and *bookings* to follow the *booking* rather than those it applies to and those it doesn't.
Following Bookings	

© 2010-2015 Exprodo Software

6.6.3.4.6 Number Of Bookings Rule

The **Number of Bookings Rule** allows a limit on the total number of *bookings* [446] that can be made. For example, the number of approved/requested *bookings* can be limited for each *project* [448]. This can be a fixed limit, or the limit can be selected from the *project's* settings.

The number of *bookings* can be specified in the *Booking Rule* [446] or a value can be read from a *property* [448], so research *projects* can be set up to have a limited number of *bookings* defined in their *project* definition, but it can be specified that *bookings* on the weekend do not count towards that total using **Bookings to Ignore**.



Property	Description
Reject or warn when exceeded	Decide whether to reject the *booking* or just post a warning.
Retry approved bookings	An indication to be sent to the user of whether to retry sending the *booking* as a **Request** (only applicable when the *booking* received is **Approved**).
Number Bookings	Set the number of *bookings* to be a specified **Fixed Value** or a **Variable Value**, something that is read from a *property* accessed through the booking. A negative number means unlimited.
Message to show	The message to be displayed to the user if the *booking* is rejected or a warning is issued.
Bookings to count	The **Bookings to Count** and **Bookings to Ignore** work in exactly the same way as *Applies To* [446] and *Does Not Apply To* [447], read the chapter on Choosing Which Bookings A Rule Applies To [226] for more information on how the sub tabs work. The main change is that there is a **Matched** option for users, *projects*, *resources* [449], *user types* [450], *project types* [448] and *resource types* [449]. This specifies that the *bookings* to be counted/ignored must be compared with the current *booking* and match this *booking* in some way, either by user, *project* or *resource* (or type of each). For example setting **Bookers** equals *Matched* **User** and **Resource** to **Matched Resource** means *bookings* for this user on the current *resource* only, will be looked at when counting *bookings* to count/ignore, which allows setting up very generic *Booking Rules* without needing a similar *Booking Rule* for every user in the system.
Bookings to ignore	

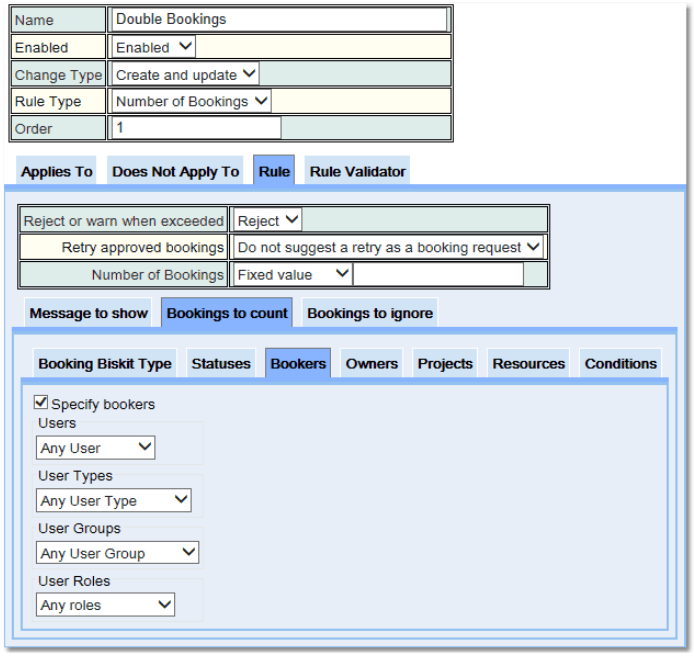
© 2010-2015 Exprodo Software

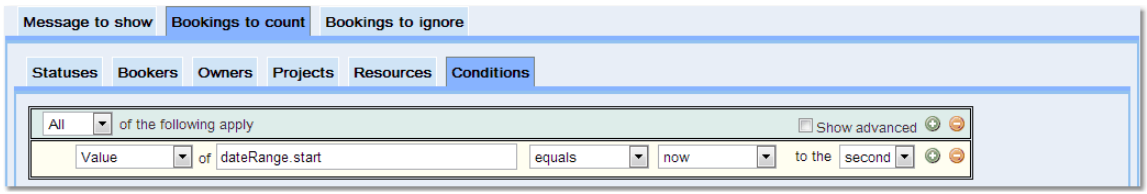
Example: Allowing a user to only have a certain number of future bookings at any one time

The facility may want to limit users to a specific number of *bookings* at any one time (or in a particular time period). Once a *booking* is used the user can then make another *booking*. As an example a user may only have three future *bookings* on a *resource* at any one time.

To do this create a **Number of Bookings Rule**, give it a name and set the **Number of Bookings** to be 3. (Under the **Rule** tab). Then set up some *Booking Rules*, the first part will count *bookings* matching the current user, the second part will count only future *bookings*. Use **New value** rather than **Old value** so that already created *bookings* that are moved from somewhere where they are not counted, to somewhere where they will be counted will be caught.

Section	Tab	Description
Bookings to count	**Bookers**	**Users** set to **Matched User**
Bookings to count	**Conditions**	**Value** of **dateRange.start** later than now to the second





The following is set up by default, defining that the *bookings* must be for the same *resource*.

Section	Tab	Description
Bookings to count	**Resources**	**Resources** set to **Matched Resource**

© 2010-2015 Exprodo Software

This could be expanded to *resource types* ensuring a user could book no more than three times over a number of *resources* of the same type or removed altogether to restrict a user to a maximum of three *bookings* over the whole facility.

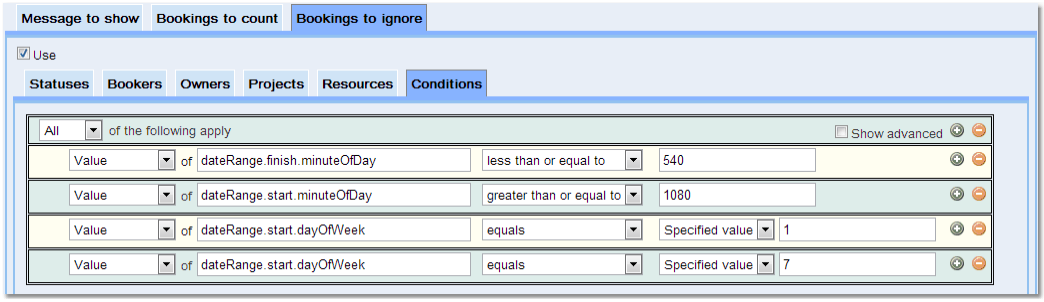
To limit the period to the next three weeks set up an additional *condition*[447]:

Section	Tab	Description
Bookings to count	**Conditions**	**Value** of **dateRange.start** no later than now plus 21 days to the second

This *Booking Rule* could be on a per *project* basis rather than a per *booker*[446] basis, just use **Projects->Matched Project** instead of **Bookers->Matched User**.

To ignore *bookings* finishing before 9AM, starting after 6PM or on a Saturday or Sunday:

Section	Tab	Description
Bookings to ignore	**Conditions**	**Value** of **dateRange.finish.minuteOfDay** less than or equal to 540
Bookings to ignore	**Conditions**	**Value** of **dateRange.start.minuteOfDay** greater than or equal to 1080
Bookings to ignore	**Conditions**	**Value** of **dateRange.start.dayOfWeek** equals **Specified Value** 1 (Sunday)
Bookings to ignore	**Conditions**	**Value** of **dateRange.start.dayOfWeek** equals **Specified Value** 7 (Saturday)



Please note the change to **Any of the following apply.**

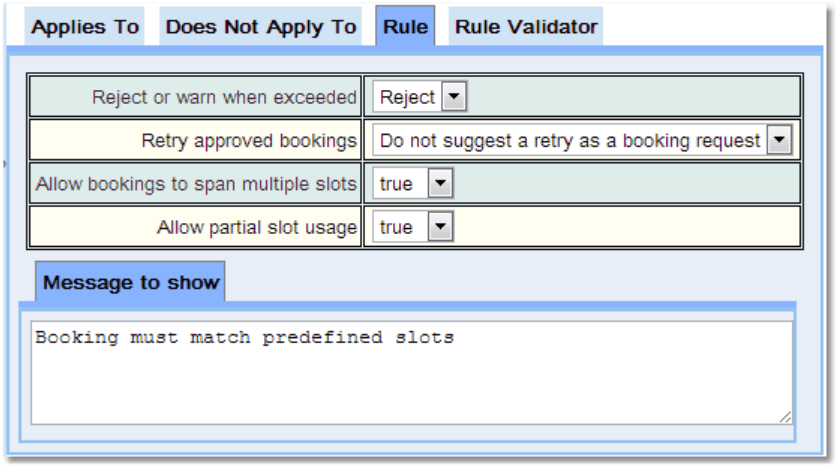
Do something similar in **Does Not Apply To**. This will make sure that when creating the *booking* it is ignored in the count.

Section	Tab	Description
Does Not Apply To	**Conditions**	**New value** of **dateRange.finish.minuteOfDay** less than or equal to 540
Does Not Apply To	**Conditions**	**New value** of **dateRange.start.minuteOfDay** greater than or equal to 1080
Does Not Apply To	**Conditions**	**New value** of **dateRange.start.dayOfWeek** equals **Specified Value** 1 (Sunday)
Does Not Apply To	**Conditions**	**New value** of **dateRange.start.dayOfWeek** equals **Specified Value** 7 (Saturday)

© 2010-2015 Exprodo Software

6.6.3.4.7 Predefined Slots Booking Rule

The **Predefined Slots Booking Rule** allows the administrator to specify the times of the day that *bookings* [446] must adhere to. This can be used to enforce time slots starting and finishing at particular times. The *Booking Rule* [446] will also enforce whether *bookings* may go over multiple time slots or partial time slots may be booked. The time slots for this *Booking Rule* are set up in the Resource Editor [270].



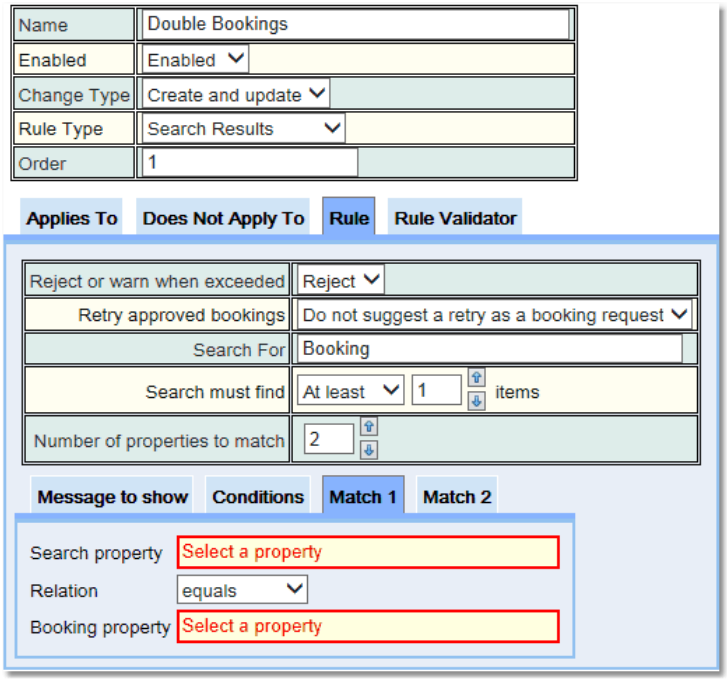
Property	Description
Reject or warn when exceeded	Decide whether to reject the *booking* or just post a warning.
Retry approved bookings	An indication to be sent to the user of whether to retry sending the *booking* as a **Request** (only applicable when the *booking* received is **Approved**).
Allow bookings to span multiple slots	Defines whether a *booking* can go over multiple defined slots
Allow partial slot usage	Defines whether a user can make a *booking* in the rest of a slot after someone not confined by the slots (such as **Admin**) has made a *booking* in only part of a slot.
Message to show	The message to be displayed to the user if the *booking* is rejected or a warning is issued.

© 2010-2015 Exprodo Software

6.6.3.4.8 Search Results Booking Rule

The **Search Results Booking Rule** allows the administrator to check for particular information on a different *Biskit Type*[446] before allowing the *booking*[446] to go ahead. One possible use of this *Booking Rule*[446] is to check whether a *booker*[446] has been trained on a particular *resource*[449]. If the administrator has created a *Biskit Type* to hold training information, such as date, *resource*, trainee name, then the *Booking Rule* can check whether a potential *booker* is trained.

The *Booking Rule* searches for a *Biskit Type*, the administrator can then specify how many *properties*[448] in the *booking* and the *Biskit Type* must match, and the names of those *properties and* how many record matches are required. As with other *Booking Rules* additional *conditions*[447] can be set up. I.e. training, the *Booking Rule* could match the trainee name with the *bookers* name, the *resource* name with the training *resource* name, and have a condition to check whether the training was done within the last year.



Property	Description
Reject or warn when exceeded	Decide whether to reject the *booking* or just post a warning.
Retry approved bookings	An indication to be sent to the user of whether to retry sending the booking as a **Request** (only applicable when the *booking* received is **Approved**).
Search For	The *Biskit Type* to compare against
Search must find	How many records must be found to match
Number of properties to match	How many *properties* will need to have their values matched between the *booking* and the searched for *Biskit Type*. For each one required a **Match** option will appear below.
Message to show	The message to be displayed to the user if the *booking* is rejected or a warning is issued.
Match ?	The names of the *properties* of the *booking* and the **Search** *Biskit Type* to be compared and the type of comparison.

© 2010-2015 Exprodo Software

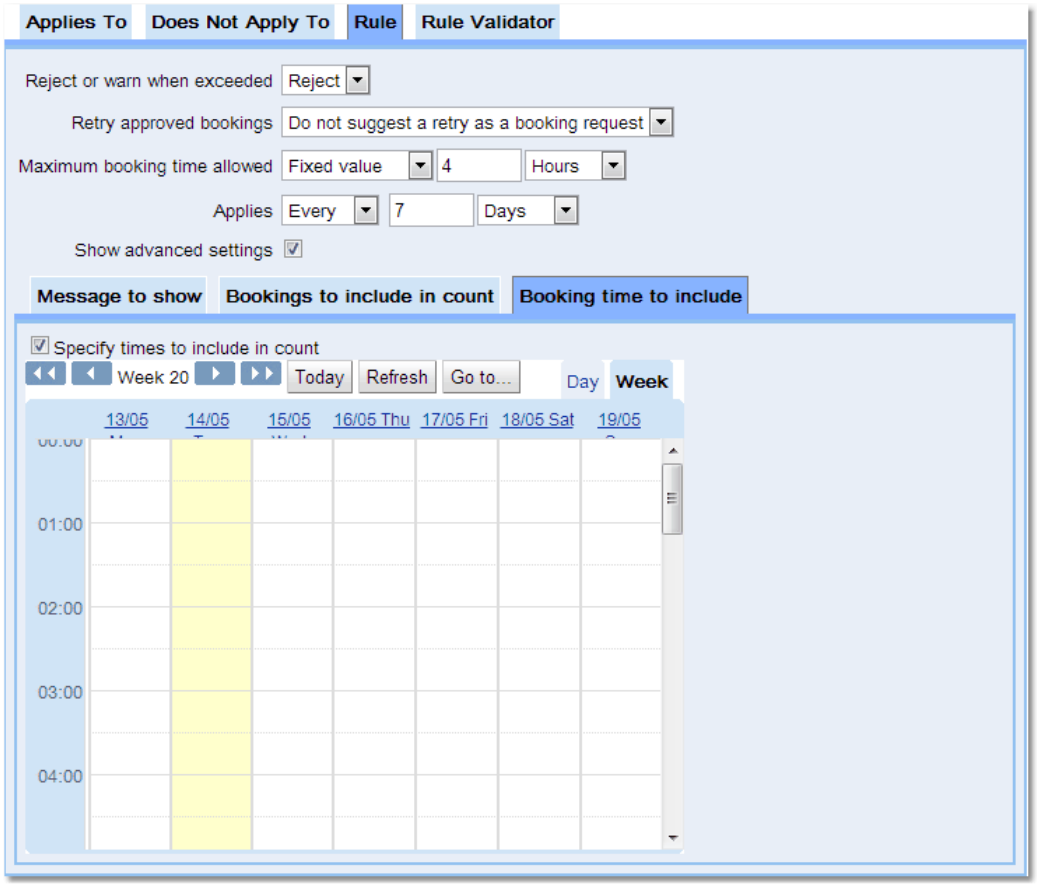
6.6.3.4.9 Total Time Booked Rule

A **Total Time Booked Rule** checks how much time is being booked over a given period of time. For example, the amount of time booked per *project*[448] or user within any 7 day period can be limited. A *Booking Rule*[446] like this may need to be implemented if the *resources*[449] are extremely busy, and it is required to restrict the use an individual has of the *resources* so that everybody has a chance to get their fair share of them.

It could also be decided that it is necessary to limit how much time is used during some core operating hours, while allowing out-of-hours usage not to count towards the usage limit. This can then encourage people to use your *resources* during less busy periods. This can be done either day-by-day, or during whatever time periods you want. For example, there may be a **Total Time Booked Rule** that limits the time booked from 8am to 6pm Monday to Friday, while having another *Booking Rule* that puts a higher limit on usage from 6am to 8pm Monday to Friday, thereby encouraging use earlier and later in the day.

> **Note**
>
> If you limit usage to 4 hours in any 7 day period, and somebody has a *booking* for 4 hours on a Tuesday afternoon, and another *booking* for 4 hours on the following Tuesday morning, this will count as 8 hours use in 7 days, contravening the 4 hour limit.

For this to work, you need to tell **Calpendo**:



© 2010-2015 Exprodo Software

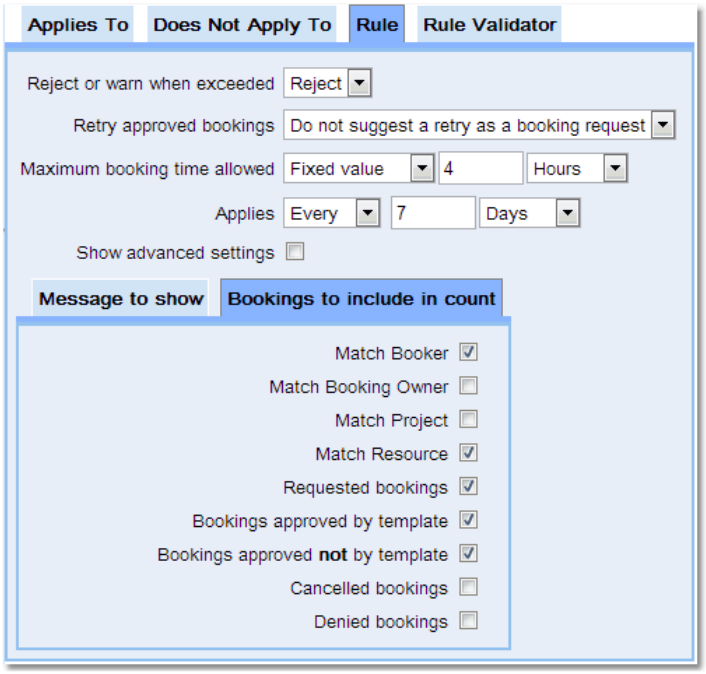
Name	Default Value	Description
Reject or warn when exceeded	Reject	Whether to reject or warn for *booking* too much time.
Retry approved bookings	Do not suggest a retry as a booking request	Whether *approved bookings* 446 should be tried again as a request.
Maximum booking time allowed	4 hours	The total amount of time that is allowed to be booked. This can be a fixed value or read from a *property* 448. A negative number means unlimited.
Applies	Every 7 days	The time period over which users can book the **Maximum booking time allowed**. This can be over a time period (**Every**) or in total (**In Total**)
Message to show	There can be a total of [hoursAllowed] hours and [minutesAllowed] minutes, but this booking would make a total of [hoursUsed] hours and [minutesUsed] minutes.	This is the message to be shown to the user, and this can include dynamically-generated text that indicates the amount of time allowed to be booked and the time used, if the requested *booking* were allowed.
Bookings To Include In Count	Requested, Approved and Approved by Template, Matched Booker and Matched Resource	Specifies which *bookings* should be counted towards the amount of the **Maximum booking time allowed**.
Bookings Time To Include	All time	Indicates which periods of time should count towards the amount of the **Maximum booking time allowed**. For example, should we only include time booked from 8am to 6pm?

© 2010-2015 Exprodo Software

Bookings To Include In Count

The **Total Time Booked Rule** is triggered when somebody tries to create or update a *booking*. When looking for other *bookings* to include in the count towards the amount of time used, specify *bookings* using the same tabs that appear in the **Applies To** and **Does Not Apply To** sections of every *Booking Rule*. However, by default, a simpler interface is shown. This chooses whether to count *bookings*:

- with the same *booker*[446]
- with the same *owner*[448]
- with the same *project*[448]
- with the same *resource*[449]
- of each possible *booking status*[447]



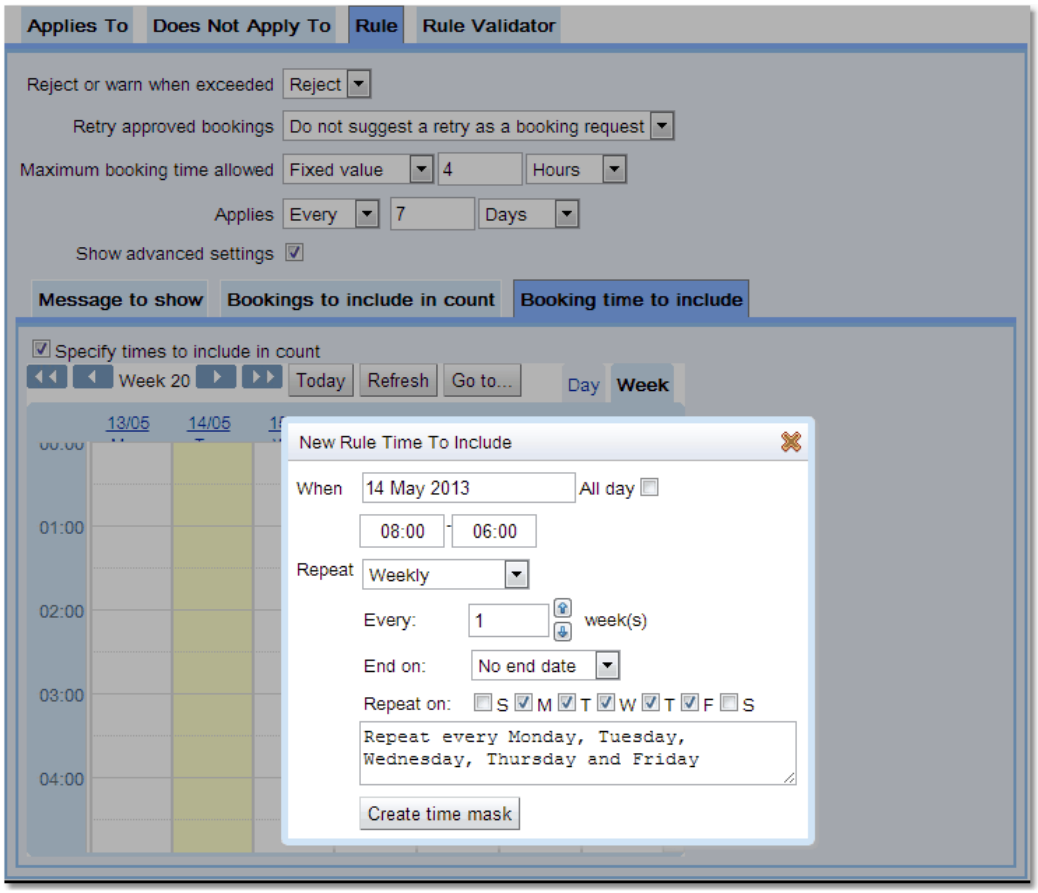
Tick the **Show Advanced Settings** check box if it is required to use the extended options that include the extra tabs. The advanced **Bookings to include in count** works in exactly the same way as *Applies To*[446], read the chapter on Choosing Which Bookings A Rule Applies To[226] for more information on how the sub tabs work.

© 2010-2015 Exprodo Software

Booking Time To Include

The **Booking Time To Include** tab does not show unless the **Show Advanced Settings** checkbox is ticked. Without this, all time booked within the **Applies To** are included in the count. To choose the periods of time that should count towards the total, tick **Show Advanced Settings** and go to the **Bookings Time To Include** tab.

If a new *Booking Rule* is being created, then the *Booking Rule* must be saved first, and then go back into edit mode in order to specify the times that should be included in the count. A calendar display will appear allows the creation of entries very similar to the way that *bookings* are created on the *Bookings Calendar*[446]. This is the way to specify the times that should be included in the count - if there's an entry in the **Bookings Time To Include** calendar, then the time will count.



As an example to make a **Total Time Booked Rule** count periods of time that are from 8am to 6pm, from Monday to Friday, create an entry for Mon-Fri that *repeats*[449] weekly. For more information on setting up *repeat* time masks read the section on Repeat Bookings. Once the times are selected click the **Create time Mask** button. There can be multiple time masks on the calendar.

© 2010-2015 Exprodo Software

Example: Allowing a user to only have a certain number of hours of future bookings at any one time

The facility may want to limit users to having a specific number of hours of *bookings* at any one time (or in a particular time period). Once a *booking* is used the user can then make another *booking*. As an example a user may only have four hours of future *bookings* on a *resource* at any one time.

To do this create a **Total Time Booked Rule**, give it a name. Set the **Maximum booking time allowed** to four and **Applies** to **in Total**. (Both under the **Rule** tab)

Section	Tab	Description
Bookings to include in count	**Conditions**	**Value** of **dateRange.start** later than now to the second

The following is set up by default in this *Booking Rule*, which defines that the *bookings* must be for the same *booker* and the same *resource*.

Section	Tab	Description
Bookings to include in count	**Bookers**	**Users** set to **Matched User**
Bookings to include in count	**Resources**	**Resources** set to **Matched Resource**

of course this could be expanded to either *resource groups* 449 or *resource types* 449 ensuring a user could not book more than four hours over a number of *resources* or removed altogether to restrict a user to a maximum of four hours over the whole facility.

Read the example for the Number of Bookings Rule 247 for limiting the *Booking Rule* to the next three weeks or making it *project* based rather than user based.

Use the calendar in Booking Time To Include 254 to set those hours that will be counted towards the total.

© 2010-2015 Exprodo Software

6.6.3.4.10 Advanced Booking Rule

Advanced Booking Rules provide a means for the implementation of almost anything required. They allow the configurer to write a *Booking Rule* using Java that will be run using BeanShell. This means it requires a programmer to create an **Advanced Booking Rule**. This documentation includes some example *Booking Rules*[446], so that a programmer may be able to produce the effect they want from what's here, but this is not a complete reference to the functionality available inside an **Advanced Booking Rule**. A complete programmer's API documentation may be made available at some time, but for now, to create an **Advanced Booking Rule**, either try to adapt the examples here, or contact us for help.

When creating an Advanced Rule a user needs to define how Advanced Rules will work with *repeat bookings*[449]. Normally rules are run once for each instance of a *repeat booking.*

Option	Description
Run for each repeating instance	The *Rule*[446] is given a *copy* of the *repeat booking*. Consequently, if the rule modifies this copy, changes made to it will not be saved to the database nor seen by subsequently run *rules*.
Run once	The *rule* is given the original repeat *booking*, and changes the *Rule* makes to the *booking* will then be persisted and also available to subsequent *Rules*.

Note: that for *repeat bookings*, advanced *Rules* marked as being run once will always be run before all other *Rules* regardless of requested *Rule* execution order.

© 2010-2015 Exprodo Software

When your *Booking Rule* is run, the following local variables will be set in the *BeanShell* environment:

Name	Class	Description
biskitDAO	com.springsolutions.biskit.persistence.HibernateBiskitDAO	Interface to the **Calpendo** database.
biskitDefs	com.springsolutions.biskit.core.def.BiskitDefStore	Provides access to all the *Biskit*[446] definitions.
booking	com.springsolutions.calpendo.domain.Booking	The *booking*[446] being created or the new version of the *booking* being updated. Identical to **newBooking**.
newBooking	com.springsolutions.calpendo.domain.Booking	The *booking* being created or the new version of the *booking* being updated. Identical to **booking**.
oldBooking	com.springsolutions.calpendo.domain.Booking	The original version of the *booking* for an update, and **null** otherwise.
realUser	com.springsolutions.calpendo.domain.CalpendoUser	The user making the change.
user	com.springsolutions.calpendo.domain.CalpendoUser	The effective user making the change. This can be different from the real user when running with The Booking Rule Validator[264].
isUpdate	Boolean	**True** if the user is making an update, and **false** if a *booking* is being created.
result	com.springsolutions.calpendo.domain.rule.RulesResult	The result that must be filled in by the *Booking Rule*.
rule	com.springsolutions.calpendo.domain.rule.AdvancedRule	The *Booking Rule* itself. Sometimes you may want to insert the *Booking Rule* name into the error message shown to a user.

© 2010-2015 Exprodo Software

API

The **RulesResult** class, as represented by local variable **result**, is the way to pass information back to **Calpendo**. It has the following API:

```
public interface RulesResult
{
  /** Sets the message that should be sent to the user. */
  public void setMessage(String msg);

  /** Sets an output string to be displayed when calling via
    * The Rule Validator. */
  public void setOutput(String output);

  /** Specify whether to reject, warn or accept the booking. */
  public void setRejectionLevel(RejectionLevel level);

  /** Specify whether to retry the booking as a request.
    * This is ignored if the booking status was not Approved. */
  public void setTryBookingRequest(boolean tryRequest);
}
```

RejectionLevel is an enum as follows:

```
package com.springsolutions.calpendo.domain.rule;

public enum RejectionLevel
{
   ACCEPT("Accept booking"),
   WARN("Warn"),
   REJECT("Reject");

   private String m_label;

   private RejectionLevel(String label)
   {
      m_label = label;
   }

   @Override
   public String toString()
   {
      return m_label;
   }
}
```

© 2010-2015 Exprodo Software

Example: Reject Bookings More Than 6 Weeks Into The Future

Here is an example of an **Advanced Booking Rule** that will reject *bookings* made more than 6 weeks into the future. Note that this can be done both with a **Simple Booking Rule** and also (better still) with *Time Templates* 450. However, this serves as an example of how to use advanced *Booking Rules*.

```
import com.springsolutions.calpendo.domain.*;
import com.springsolutions.calpendo.domain.rule.*;
import java.util.Calendar;

// Throw away daylight savings effect.
// This is technically incorrect, but is what people expect when
// crossing DST boundaries (ie use the time shown on the wall
// clock, not the number of minutes between now and then)

then_cal = Calendar.getInstance();
then_cal.setTime(booking.getDateRange().getStart());
then_cal.set(Calendar.DST_OFFSET, 0);
then_time = then_cal.getTime().getTime();

now_cal = Calendar.getInstance();
now_cal.set(Calendar.DST_OFFSET, 0);
now_time = now_cal.getTime().getTime();

days = (then_time - now_time)/(1000.0*60*60*24);

if (days > 42)
{
  int idays = (int) days;
  int ihours = (int) ((days-idays)*24);
  int imins = (int) ((days - idays)*24*60 - ihours*60);
  result.setRejectionLevel(RejectionLevel.REJECT);
  result.setMessage(
        "You cannot book more than 6 weeks in advance "
        + "(your request was " + idays + " days, "
        + ihours + " hours and " + imins
        + " minutes in advance)."
        + " Please discuss any special requirements with a "
        + "moderator. A list of moderators can be found under "
        + "the search menu." );
}
```

© 2010-2015 Exprodo Software

Example: Enforcing Project Resource Settings

Calpendo uses Hibernate for accessing its database. The database can be accessed directly by obtaining a Hibernate SessionFactory as shown by this example. If this is done, make sure that the Hibernate session is closed. This *Booking Rule* does the following:

- Only run for *bookings* that have a *project*[448] associated
- Reject the *booking* if trying to book for a *resource*[449] that's not represented by the *Project's Resource Settings*[448]
- Reject the *booking* if it is for a time after the *project's* **finish** *property*[448], where **finish** is a dynamic *property*.
- Reject the *booking* if its duration is more than the **minutesPerSession** *property* of the *Project's Resource Settings*
- Use Hibernate to count the total number of **Approved** and **Requested** *bookings* for the *project*, and reject the *booking* if there would be more *bookings* than the **numberOfSessions** *property* of the *Project's Resource Settings*

These tasks are more easily done using the dedicated *Booking Rule Types*[446], but this *Booking Rule* serves as a good example of how to interact with the **Calpendo** database directly through Hibernate.

```
import com.springsolutions.calpendo.domain.*;
import com.springsolutions.calpendo.domain.rule.*;
import com.springsolutions.exprodo.core.server.HibernateUtil;
import java.util.Calendar;
import java.util.Date;
import org.hibernate.Session;
import org.hibernate.SessionFactory;

resource = booking.getResource();
rpk = resource.getPrimaryKey();
project = booking.getProject();
if (project == null) return;

prsSet = project.getResourceSettings();
prs = null;
for (ProjectResourceSettings tmp : prsSet)
{
  if (tmp.getResource().getPrimaryKey() == rpk)
  {
    prs = tmp;
    break;
  }
}

if (prs == null)
{
  result.setRejectionLevel(RejectionLevel.REJECT);
  result.setMessage(
    "You are trying to book for a project ("
    + project.getProjectCode()
    + ", " + project.getName()
    + ") that has not been approved for using "
```

© 2010-2015 Exprodo Software

```
    + resource.getName());
  return;
}

endDateProp = project.get("finish");
endDate = (endDateProp == null) ? null : endDateProp.getValue();
if (endDate != null
    && booking.getDateRange().getFinish().after(endDate))
{
  result.setRejectionLevel(RejectionLevel.REJECT);
  result.setMessage(
    "You are trying to book for a project ("
    + project.getProjectCode()
    + ", " + project.getName()
    + ") after its project end date (" + endDate + ")");
  return;
}

duration = booking.getDurationInMinutes();
maxDuration = prs.getMinutesPerSession();
if (duration > maxDuration)
{
  result.setRejectionLevel(RejectionLevel.REJECT);
  result.setMessage(
    "Bookings for project (" + project.getProjectCode()
    + ", " + project.getName()
    + ") should be no longer than " + maxDuration
    + " minutes, but you've tried to make a booking for "
    + duration + " minutes");
  return;
}

hql = "select count(*) from Booking where project.id = "
       + project.getPrimaryKey()
       + " and (status = 'REQUESTED' or status = 'APPROVED') "
       + " and resource.id = " + rpk;

if (booking.getPrimaryKey() != 0)
  hql = hql + " and id != " + booking.getPrimaryKey();

SessionFactory sf = HibernateUtil.getSessionFactory();
Session session = null;
try
{
  session = sf.openSession();
  count = session.createQuery(hql).uniqueResult();
  print("Query found a total of " + count
        + " bookings for this project and resource");
  if (booking.getPrimaryKey() == 0)
    count++;

  maxCount = prs.getNumberOfSessions();

  if (count > maxCount)
  {
    result.setRejectionLevel(RejectionLevel.REJECT);
    result.setMessage(
```

© 2010-2015 Exprodo Software

```
      "There can be a total of " + maxCount
      + " bookings for project (" + project.getProjectCode()
      + ", " + project.getName()
      + " on " + resource.getName()
      + "), but this booking would make a total of " + count);
    return;
  }
}
finally
{
  if (session != null)
    session.close();
}
```

Example: Enforce 15 Minute Gaps Between Bookings

This *Booking Rule* will ensure that there is always a 15 minute gap between *bookings*. If using this *Booking Rule*, it may be a good idea to use the *Rule's* **Applies To --> Resources** tab to make sure that it only *applies to*[446] the appropriate *resources*.

```
import com.springsolutions.biskit.core.Biskit;
import com.springsolutions.calpendo.domain.Booking;
import com.springsolutions.calpendo.domain.rule.RejectionLevel;
import com.springsolutions.exprodo.core.server.HibernateUtil;
import com.springsolutions.timeRepeating.domain.RepeatUtil;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import org.hibernate.Session;
import org.hibernate.Query;

session = HibernateUtil.getSessionFactory().openSession();
sqlFormatter = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

try
{
  Calendar cal = Calendar.getInstance();
  cal.setTime(booking.getDateRange().getStart());
  cal.add(Calendar.MINUTE, -15);
  Date start = cal.getTime();
  cal.setTime(booking.getDateRange().getFinish());
  cal.add(Calendar.MINUTE, +15);
  Date finish = cal.getTime();

  String hql = "from Booking b where not (b.dateRange.start > '"
          + sqlFormatter.format(finish)
          + "' or b.dateRange.finish < '"
          + sqlFormatter.format(start)
          + "') and b.resource.id="
          + booking.getResource().getPrimaryKey()
          + " and b.status != 'CANCELLED' "
          + " and b.status != 'DENIED'";

  searchResult = session.createQuery(hql).list();
```

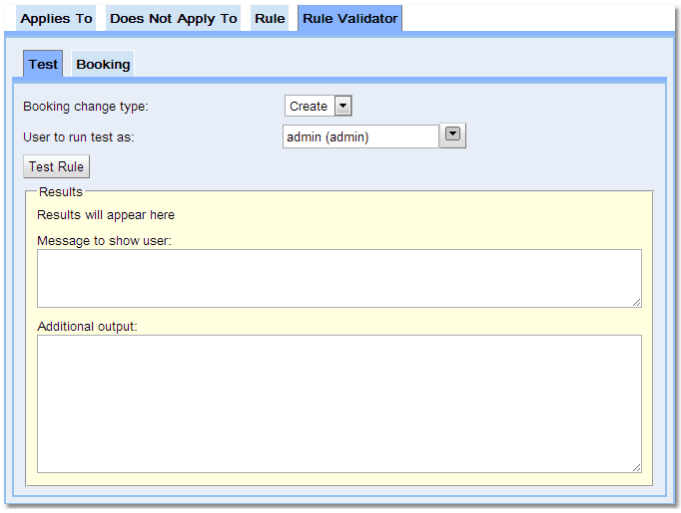
© 2010-2015 Exprodo Software

```
  found = RepeatUtil.expandRepeatables(searchResult, start, finish);
  boolean doubleBooking=false, bufferInfringed=false;
  bookingStart = booking.getDateRange().getStart().getTime();
  bookingFinish = booking.getDateRange().getFinish().getTime();

  for (Biskit b : found)
  {
    Booking clash = (Booking) b;
    dr = clash.getDateRange();
    if (b.getPrimaryKey() == booking.getPrimaryKey()
      || dr.getStart().getTime() >= finish.getTime()
      || dr.getFinish().getTime() <= start.getTime())
      continue;

    if (dr.getStart().getTime() >= bookingFinish
      || dr.getFinish().getTime() <= bookingStart)
      bufferInfringed = true;
    else
      doubleBooking = true;
  }

  String msg;
  if (doubleBooking)
    result.setMessage("Double bookings are not allowed");
  else if (bufferInfringed)
    result.setMessage(
      "You must leave a gap of 15 minutes between bookings");

  if (doubleBooking || bufferInfringed)
    result.setRejectionLevel(RejectionLevel.REJECT);
}
finally
{
  session.close();
}
```

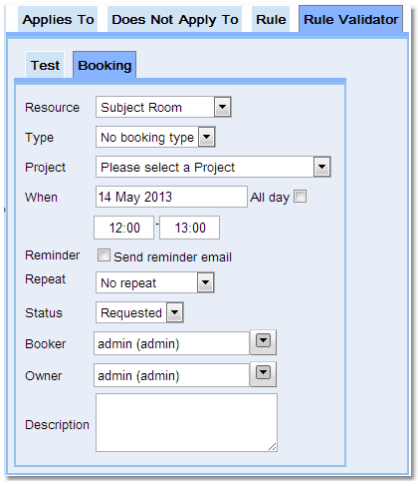
© 2010-2015 Exprodo Software

6.6.3.5 The Booking Rule Validator

The **Booking Rule Validator** tab of the **Booking Rule Editor** provides a means of testing any user created *Booking Rules*[446]. In particular, please note that when a *Booking Rule* is edited, it can be tested before it is saved, so that its behaviour can be tested to make sure it works as expected.



The way the **Booking Rule Validator** works, is to simulate creating a booking or updating a *booking* by changing the **Booking** change type. The **Booking** tab is where the values of the *booking* being created is specified, or, when simulating an update, this is replaced by two tabs **Old Booking** and **New Booking** where the old and new values for the simulated *booking* can be specified. Then decide which user the test is being run as. **Running** the test while pretending to be another user is a way to see how the *Booking Rule* would behave when a *booking* is created or update by somebody else.



Finally, the **Test Rule** button runs the test. The text area at the very bottom of the tab shows the output and the rejection level returned from the *Booking Rule* with information about what happened. If there's a message that would be presented to the user, or the *Booking Rule* provided some other output by calling RulesResult.setOutput(String)[258], then it will be shown in the **Message to show** user area.

© 2010-2015 Exprodo Software

6.7 Types And Groups

For various reasons, it is useful to be able to split users into sets of one sort or another. This is done in **Calpendo** using types and groups. These are two independent means of segregating users, projects and resources.

> **The Difference Between Types and Groups**
>
> The key thing to understand is that a user has precisely one type, but may belong to any number of groups.

Your **Calpendo** should be configured with types and groups that make sense for you. A default installation comes with no types and no groups. If your installation does use *User Types* 450, then when a user registers, they will be asked what their type is. An administrator can override that and change it when they approve the user, but the user will always have a single type. When users have types, then the type can be used for Permissions 278, Automatic Emails 297, Manual emails 172 and Booking Rules 223.

User Types are used if there is some logical separation of users that fits the notion of a user having precisely one type. For example, it may be that there are multiple departments in your company, with everybody belonging to only one department. If that were the case, then perhaps you would have as many *User Types* as there are departments.

User Groups 450 can also be used for Permissions 278, Automatic Emails 297, Manual Emails 172 and Booking Rules 223, the same as for types. However, there is greater flexibility due to the way a user can belong to multiple groups (or no groups). So, while a *user's type* may indicate their department (or whatever), groups could be created that reflect any sort of grouping that makes sense in your situation. Given this apparent flexibility in groups, one might ask what the point of types is. The answer is that it is sometimes useful to have a taxonomy in which every user has a *type*, whereas a user may belong to no groups at all.

In exactly the same way that a User has a *User Type* and may belong to *User Groups*, a *Project* 448 has a *Project Type* 448 and may belong to *Project Groups* 448, and a *Resource* 449 has a *Resource Type* 449 and may belong to *Resource Groups* 449. The same Booking Rule about having precisely one type and belonging to any number of groups applies to *projects* and *resources* in exactly the same way as it does to users. These types and groups can be used for Permissions 278, Automatic Emails 297, Manual Emails 172 and Booking Rules 223.

Bookings 446 may also have a type. If you create one or more types in the **Types and Groups Editor**, then when somebody creates a *booking*, they will be asked for its type. However, there are no *booking* groups in **Calpendo**. One might want to use a *booking type* 447 for several reasons. For example, in a clinical or mixed clinical and research environment, it may be required to categorise scanner *bookings* according to the nature of the subject of the scan. That is, to record whether they are fully healthy, walking patients, patients in a chair or patients in a bed.

If using *booking types*, then this allows the setting of different *Booking Rules* 446 or *Permissions* 448 based on the *booking type*, and the different *booking types* can also be displayed differently on the *calendar* 446 so that the *booking type* is obvious at a glance. The *Booking Type* can also be used for Permissions 278, Automatic Emails 297, Manual Emails 172 and Booking Rules 223.

© 2010-2015 Exprodo Software

6.8 Configuring Types And Groups

The **Types And Groups Editor** is used to configure the types and groups in the system. Before reading more about how to configure them, please take a look at Types And Groups 265 which describes what a type and a group is, and how an item can have exactly one type but may be a member of many groups (including zero).

When it comes to configuring types and groups in **Calpendo**, consider first whether there is a good way to categorise the users, *projects* 448 and *resources* 449 such that they each fall into precisely one category.

For example, distinguish between students, staff and external users using that for the *user type* 450. Alternatively if multiple departments use the *resources*, its possible to assign a *user type* according to which department a user belongs to.

Project types 448 could be assigned according to the department they originated from, or perhaps according to thier funding source.

Resource types 449 tend to be more obvious, but are just as important. Categorise *resources* as Room, Scanner, Person, Computer, Desk, etc.

Once the types have been chosen, then they can be used throughout **Calpendo**. For example, to apply a particular *Booking Rule* 446 to Scanner *bookings* 446 for users of a particular type, or for *projects* 448 of a particular type. Types are used to control Booking Rules 223, Time Templates 215, Permissions 278, Automatic Emails 297 and Manual Emails 172.

Types And Groups Editor

The **Types and Groups Editor** page consists of three parts:

1. The menu tool bar: this is the tool bar at the top of the page, and all the buttons operate on all types and groups at a time.
2. The left pane: this allows the selection of a type or group or an item within a type or group
3. The right pane: this shows the detail of the currently selected type or group item

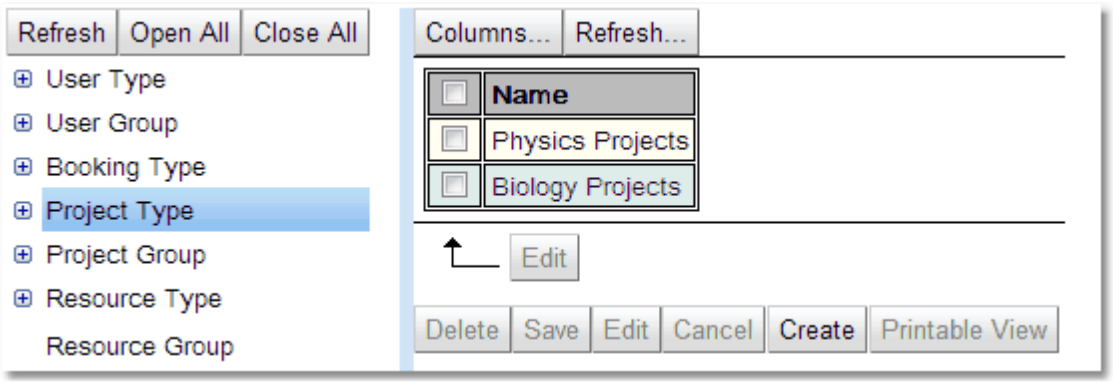
For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition 96 chapter.

© 2010-2015 Exprodo Software

The List View

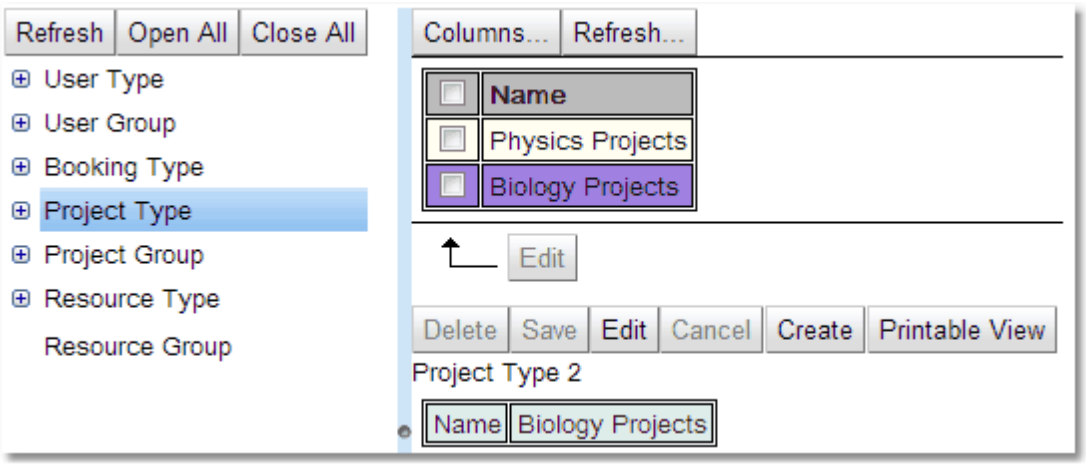
By clicking on one of the types or groups in the left hand pane, in the right hand pane will appear a list of all the entries for that type or group. This will include two menus, one above and one below the list.

For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition 96 chapter.



Many items can be edited in one go quite easily in the list view, using the check boxes read the section on How To Edit Multiple Items At Once 128 in the Data Explorer 127 chapter.

To edit an individual item from this list click on it and it will appear below the list and the edit functions allowed will no longer be greyed out.



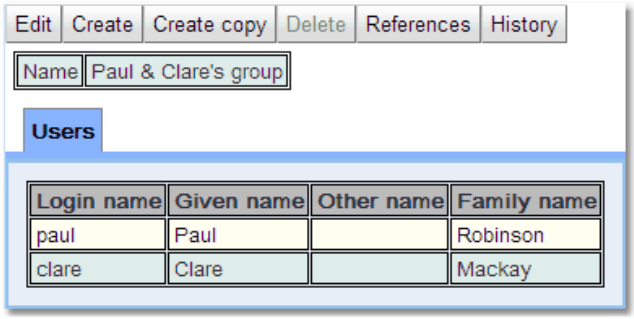
© 2010-2015 Exprodo Software

The Item View

Once a group or type has been opened up in the left hand pane select individual items to work on. Once selected these items will appear in the right hand pane.

For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition[96] chapter.

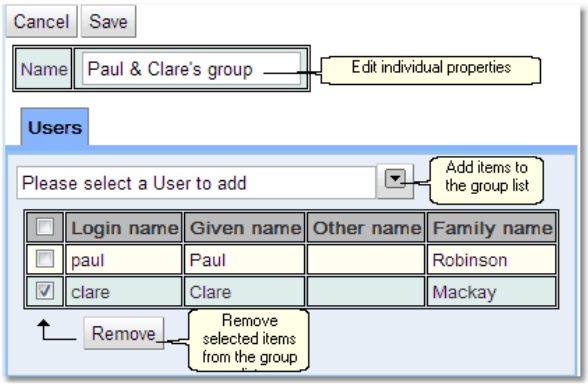
Booking Type[447]



User Group[450]

Types and Groups appear slightly differently, Types are just a list of the *properties*[448] that can be changed. A group has that list but also a list of all the appropriate items that are in its group. For instance a **User Group** will have a list of all the logins associated with that group.

Once in edit mode, edit any *properties* with *permission*[448] to access and add or delete items from group lists.

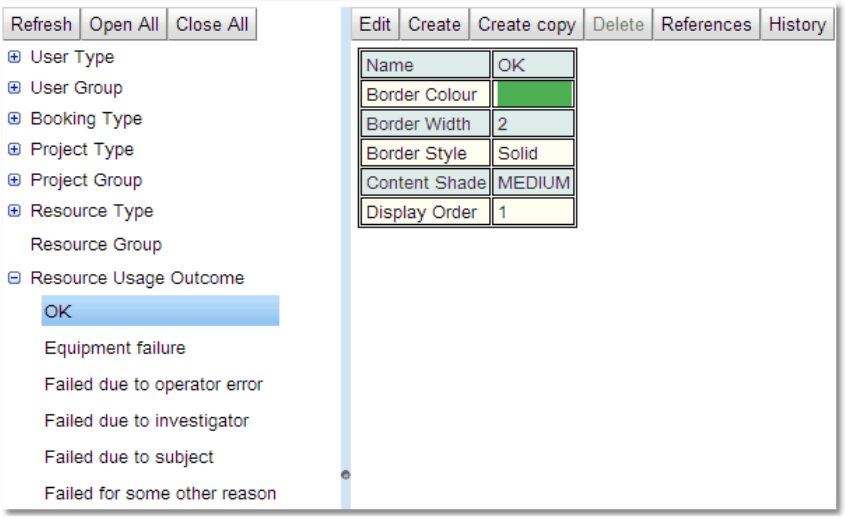


To remove items from the list first select the items to be removed using the check box and click the **Remove** button.
To add items to the list use the pull down to get a full list of possible items and select the one required.

© 2010-2015 Exprodo Software

Resource Usage Outcome

The **Types And Groups Editor** is also used to configure the outcomes that may be recorded when **Calpendo** is configured to collect *resource usage* [449] information.



Calpendo comes with some default outcomes, the administrator can add as many additional ones as required as well as configuring the choice of colour and borders for each outcome.

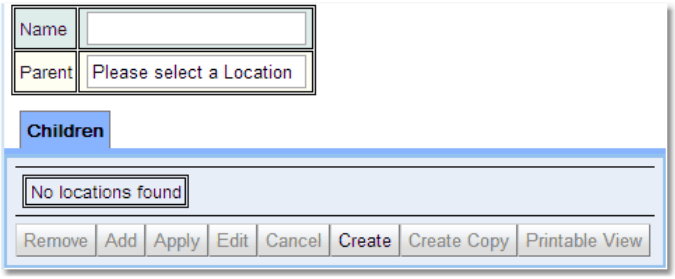
© 2010-2015 Exprodo Software

6.9 Configuring Resources And Locations

Resources[449] and **Locations** are configured in the **Resource Editor** which is, by default, available on the menu as **Admin-->Resource Editor**.

Locations

Locations have a name and a parent, this allows a hierarchy of locations to be created.

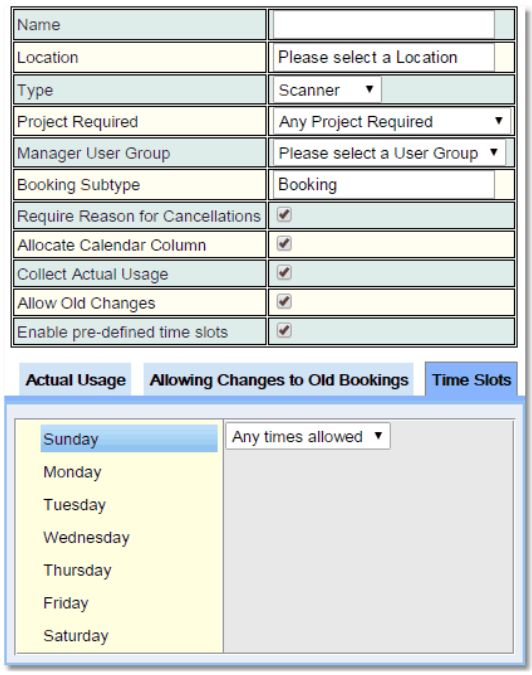


Also once a location is created the child locations can be created directly using the **Create** button in the **Children** tab.

© 2010-2015 Exprodo Software

Resources

When you create a *resource* you get the following pop up:



Resources have the following properties:

Property	Description
Name	This is the name of the *resource*. It can be anything required, except that it should be unique and it is better if the name is not very long so that its display doesn't take up much space.
Location	It is not necessary to set the location of a *resource*, but there are many *resources*, it may help to specify the location of each. The location can be used with Permissions[278] and Automatic Emails[297].
Type	*Resource types*[449] are configured using the **Types and Groups Editor** as described in Configuring Types And Groups[266]. *Resource types* can be used for Time Templates[215], Permissions[278] and Automatic Emails[297].
Project Required	There are three possible values: 1. **Project Not Required**, 2. **Any Project Required**, 3. **Authorised Project Required**. This allow a choice of whether *bookings*[446] for this *resource* must specify a *project*[448]. Typically,there might be different types of *resource* and *projects* are only required for some. For example, *projects* may need to be associated with *bookings* for an MRI scanner, but not for a conference room. Also there is the option of allowing *projects* to book a *resource* only if they have an entry for that *resource* in their *Project Resource Settings.*[448]

© 2010-2015 Exprodo Software

Property	Description
Manager User Group	To allow the administrator to point to a group of users that will manage this device. This property can then be used in Rules, Permissions, Automatic E-mails etc.
Booking Sub-Type	To allow the administrator to define different **Booking** *Biskit Types* 446 for different *resources*. All new **Booking** *Biskit Types* are created in the *Bakery* 446 and must inherit from **Booking**. They then may have their own list of *properties* 448, allowing the set up of different *booking* information for different *resources*
Require Reason for Cancellations	By default a user can cancel a *booking* without giving a reason, if this *property* is set to **True** then any cancellation of this *resource type* 449 will require a reason to be input.
Allocate Calendar Column	This allows the user to specify whether this *resource* should always use a column in the calendar when in **Always present** mode. Some *resources* such as **Annual Leave** may only be required to be shown in a column when someone has booked a half day or some hours off, the rest of the time as full day *bookings* they would be shown at the top of each days columns, and a column for each day is not required.
Collect Actual Usage	Indicates whether actual *resource usage* information should be collected for this *resource*.
Allow Old Changes	By default, *bookings* that are in the past or in progress cannot be modified. However, it may be necessary to create or modify past *bookings* so information relating to actual *resource usage* 449 can be recorded after the event. In order to do this set **Allow Old Changes** and also add *properties* to *Bookings* using the Bakery 351 to record any information that needs to be captured. It is also possible to specify the types of changes allowed and the time period these changes must be made in.
Enable pre-defined time slots	This enables the ability to create pre defined time slots for *bookings*. Ticking the check box enables the **Time Slots** tab where time slots can be set up for each of the *resources* for each day of the week.
Self	**Self** is always a link back to the *resource* in question. This is particularly useful for *Permissions* 448 where a *resource* needs to be hidden from some people. Use a *condition* 447 like **value of self equals 3T** where 3T is selected from a drop-down of *resources*. Previously, a *condition* like **value of name equals 3T** would have been required, where the text **3T** needed to be typed in. The *Permission* would have failed if the *resource* **3T** were ever renamed. This is not seen in the **Resource Editor** but will be available when creating *conditions*.

The **Resource Editor** shows a list of locations and *resources*. Select a location or *resource* and press the **Edit** button to modify it, or press the **Create** button to create a new one. For more information on how the **Resource Editor** works read the chapter on Configuring Types and Groups 266 as the **Resource Editor** works in the same fashion.

After creating a new *resource* go to the Bookmark Manager 168 and make sure to add the resource to the **System Bookmarks->Show All** *bookmark* 447 so that all users will have the *resource* in the default *bookmark*.

© 2010-2015 Exprodo Software

Enabling Actual Resource Usage Recording

Resource Usage is configured by the Resource Usage 344 tab in the Global Preferences 328 page, and also by settings on each *resource*. If **Collect Actual Usage** is set to true for a *resource*, then additional information will be requested, as shown below:

Setting	Description
Font size	The **Resource Usage Recorder** page is normally used on a computer that is next to the equipment whose *resource usage* is being recorded. If it is required to view the computer from across a room, then it might be preferable to enable the large font setting for that *resource*.
Access protection	Specifies the IP address of a computer that will use the **Resource Usage Recorder** page for this *resource*. This can be set to work from any address, or restrict it to one IP address.
Resource User Selectable	Allows the administrator to define whether the person creating the *resource usage* can select the user creating the *resource usage* or whether it defaults to the logged in user.
Usage Session ID Template	The **Resource Usage Recorder** page will provide a unique session ID for each *resource usage* session. However, the way that session ID is formatted can be configured, and also what goes into it. This editor allows entry of any text and selection of any number of *properties* that are stored on the *resource usage* itself. The items that might be useful are:

Property	Description
id	A unique integer assigned to the *resource usage*.
projectUsageCount	This is the number of *resource usages* recorded for the usage's *project* so far.
resourceUsageCount	This is the number of *resource usages* recorded for the usage's *resource* so far.
resource	The *resource* itself is also accessible. For example, the session ID may be required to include the *resource's* name.

The outcomes that may be recorded for each *resource usage* can also be customised. To do this, use the Types And Groups Editor. 266

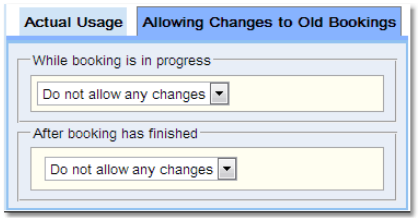
Note that there are *resource usage* menu items that remain hidden unless there is at least one *resource* with its **Collect Actual Usage** *property* to true.

© 2010-2015 Exprodo Software

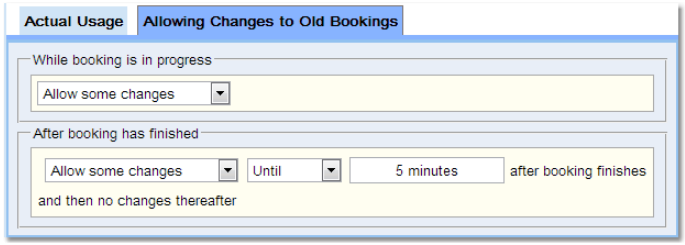
Creating Bookings In The Past, Or Enabling Changes To Past Or In Progress Bookings

To be able to create *Bookings* in the past, or allow changes to past *Bookings* or those currently in progress, set **Allow Old Changes** to **true** for a *resource*.

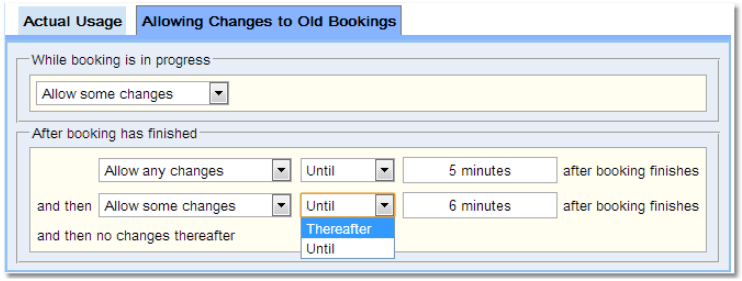
Then define whether some or all changes can be made. If **Allow some changes** is specified then the *resource, status* nor the time can be changed but all other *properties* may be (this can of course be managed by using Permissions[278]). If **Allow any changes** is specified then all *properties* may be changed unless denied by Permissions[278].


Actual Usage
Allowing Changes to Old Bookings
While booking is in progress
Do not allow any changes
After booking has finished
Do not allow any changes

If **Allow some changes** is chosen then specify whether to allow changes for a particular time period or always (Thereafter).



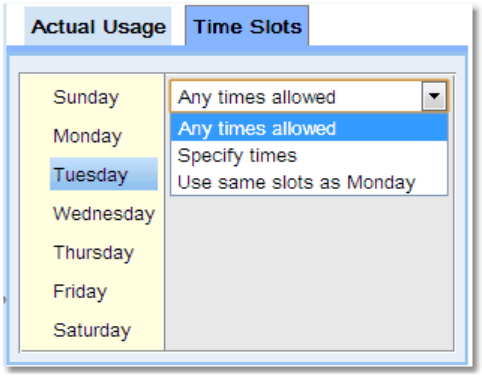
If **Allow any changes** is chosen for **After booking has finished** it can be specified whether to allow them for a particular time period or always, but it can also be specified whether to **Allow some changes** after that time. All times are taken from the time the *booking* finishes. It will not be possible to put in a shorter time into the **Allow some changes** option than is in the **Allow any changes** option.



© 2010-2015 Exprodo Software

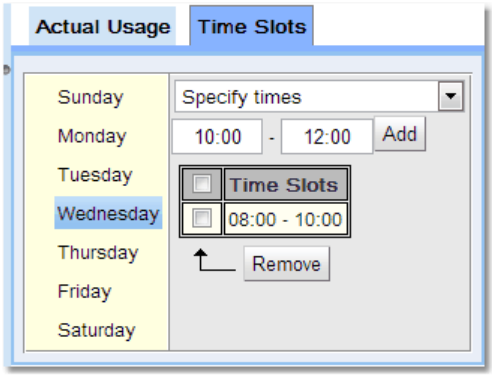
Enabling Predefined Time Slots For Bookings

If the **Enable predefined time slots** box is checked then the **Time Slots** tab can be viewed.



For each day the time slots available can be defined.

Setting	Description
Any times allowed	This is the normal operation where anyone can book any time slot.
Specify times	Define any time slots required.
Use same slots as ???day	Use the time slots defined for a different day. If that days time slots are changed then they will change on this day as well, with the exception that if the day that is being copied is changed to **Any times allowed**, then the current day will keep the slots defined before that happened and its option will change to **Specify times**.



To specify the times choose the start time and end time, and press the **Add** button. Once the first one is created Calpendo will set up the times for the next slot to be of the same duration and starting at the end of the current time slot, so if that is what is required just press **Add** again. Click the check box next to a slot and use the **Remove** button to delete any slots that are no longer required. The final slot of the day will end at 23:59 not 24:00.

Also required is that that the **Predefined Slots Booking Rule** (**Predefined slots must be adhered to** is the default Rule) is enabled.

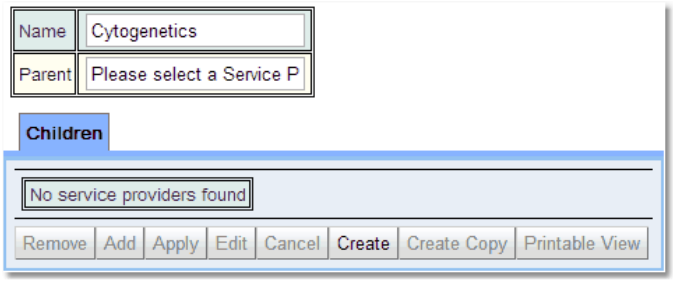
© 2010-2015 Exprodo Software

6.10 Configuring Services and Service Providers

Service Providers [449] and *Services* [449] are configured in the **Service Manager** which is, by default, available on the menu as **Admin-->Service Manager**.

Service Providers

Service Providers have a name and a parent, this allows a hierarchy of *Service Providers* to be created.



Also once a *Service Provider* is created the child *Service Providers* can be created directly using the **Create** button in the **Children** tab.

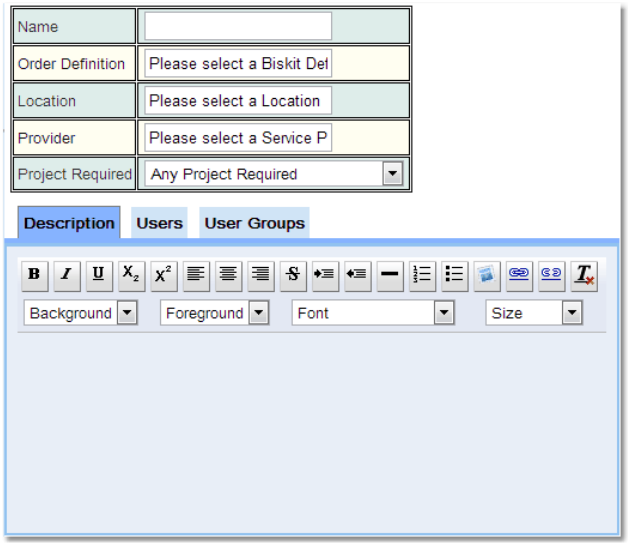
Service Orders

Service Orders [449] are created in the *Bakery* [446] by creating a *Biskit Type* [446] which inherits from the *Service Order Biskit Type,* and has all the *properties* [448] defined to create the *Service Order* and monitor it. Create one *Service Order Biskit Type* for each type of *Service* that will be required.

© 2010-2015 Exprodo Software

Services

When you configure a *Service* you get the following pop up:



Services have at least the following *properties*[448]:

Property	Description
Name	This is the name of the *service*. It can be anything required, except that it should be unique and it is better if the name is not very long so that its display doesn't take up much space.
Order Definition	The *Service Order Biskit Type* to be used to make an order for this *service*. This allows the administrator to set up different *properties* to be input for each *service* that could be ordered.
Location	It is not necessary to set the location of a *service*, but there are many *services*, it may help to specify the location of each. The location can be used with Permissions[278] and Automatic Emails[297].
Provider	The name of the *service provider*.
Project Required	There are three possible values: 1. **Project Not Required**, 2. **Any Project Required**, 3. **Authorised Project Required**. This allow a choice of whether a requester of this service must specify a *project*[448]. Typically,there might be different types of *services* and *projects* are only required for some. Option three allows *projects* to book a *service* only if they have an entry for that *service* in their *project service settings*[448].
Description	A full description of the *service*.
Users	Those users that will be emailed when an *order* is made for this *service*.
User Groups	*Groups*[450] of users that will be emailed when an *order* is made for this *service*.

© 2010-2015 Exprodo Software

6.11 Permissions

Permissions[448] are designed to be very flexible, so that you have fine-grained control over who can perform what action on what data. This can be done by creating multiple *Permissions* objects. **Calpendo** looks at all the *Permissions* that apply to the action the user is trying to perform and uses them to work out if Permission should be granted or denied.

6.11.1 How Permissions Work

Some systems have a large master list of actions, and then for each role, there would be a tick against each of the actions that users with that role can do.

The problem with that kind of approach is that it is best employed when the system has a predefined set of actions that need to be controlled. However, **Calpendo** is designed to allow for precise control over the detail of what people are allowed to do. This means that it is much more flexible for a variety of situations. That flexibility also leads to a degree of complexity, but this section of the manual will try to make this topic as simple as possible.

Permissions[448] are not implemented with an all-encompassing list of actions. The method that is used is to define a small set of simple *actions*[448], and then create customised and highly targeted individual *Permissions* to control what can be done. The overall picture is then obtained by aggregating those individual *Permissions*.

That aggregation is possible due to a mechanism that allows layering of *Permissions*, where some *Permissions* can override others. The crucial element in allowing this is the concept that **Calpendo** supports both authorising *Permissions* that permit something to be done, and non-authorising *Permissions* that deny the right to do something.

© 2010-2015 Exprodo Software

Breaking An Activity Into Its Parts

Let's start by looking at an example, viewing the content of a *booking*[446].

This breaks down into the following items:

1. ***Action***

   Start by looking at the underlying *action*. Everything is broken down into an *action* of some sort, and the most common *actions* are create, read, update and delete. In the example, it is view, or read.

2. ***Biskit Type***[446]

   Next, look at the type of the data the *action* will affect. In the example in order to control viewing of the content of a *booking*, the *Biskit Type* is *booking*.

3. ***Property***[448]

   When trying to read or update something, as in the example, then the *action* has an impact on individual *properties*. That means the user may have *permission* to read some of the *properties* on the *booking*, but not all of them.

   So for some *Permissions*, it is necessary to define the *property* controlled by the *Permission*. For example, to grant or deny the ability to view a particular *property* on a booking, create a *Permission* that affects just that one *property*.

   Permissions that apply to individual *properties* are only relevant to reading and updating things. For example, when deleting something, the whole thing is deleted and not just individual *properties*. So it makes no sense to specify a *property* on a *Permission* that controls who can delete data.

Layering Permissions

When defining *Permissions*, specify the *action* to be controlled and optionally the *Biskit Type* and *property*. *Permissions* that specify only the *action* are known as *action Permissions*. *Permissions* that specify the *action* and a *Biskit Type* are known as *Biskit Type Permissions*, and *Permissions* that specify *action, Biskit Type* and *property* are known as *property Permissions*.

In a typical system, everybody can read most things. So it is usually easier to start by giving read *Permission* to everybody to read everything. This can be done with an action *Permission* for the read action. Then, layered on top of that, create a non-authorising *Permission* that takes away *permission* for the things that are not allowed to be read.

© 2010-2015 Exprodo Software

The layering works by using *action Permissions* only if there isn't a relevant *Biskit Type* or *property Permission*. Similarly, *Biskit Type Permissions* are used if there isn't a relevant *property Permission*.

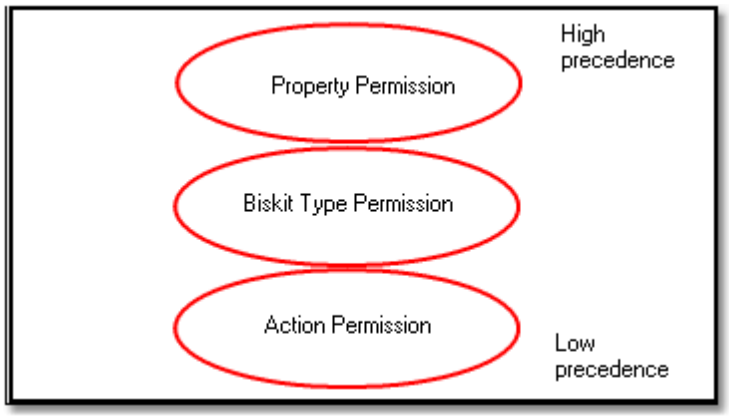
There is also a numerical priority that can be specified for each *Permission*, and that provides another way to layer *Permissions* because the highest priority *Permissions* within each of the three layers, *action*, *Biskit Type* and *property*, are considered first.

If a relevant *Permission* can not be found, then permission to perform the *action* is denied.

Precedence Of Permissions

When sorting through the *Permissions* to decide which one to apply, **Calpendo** will only consider those that match the *action*, *Biskit Type*, *properties*, *conditions* 447 and that also target the user concerned. Also, note that those *Permissions* that do not specify a *Biskit Type* will also be considered.

Once that filtering has taken place, **Calpendo** will use *property Permissions* first, then *Biskit Type Permissions*, and finally *action Permissions*. Within each of those three, *Permissions* are further sorted by priority, and the *Permission* with the highest numerical value for its priority will be used. If there are multiple *Permissions* with the same priority, then it is undefined which one will be used. The reason this is required will become apparent after going through the way in which the *conditions* under which each *Permission* may be used can be specified.



Precedence of permission types.

When a *BiskitDef* has another *BiskitDef* as its parent it will have access to the *Permissions* created for the parent.
When checking *Permissions* all the child *Permissions* will be filtered and acted on. If there is not a relevant *Permission* then the parents *Permissions* will be filtered and acted on, for each type of *Permission* in order of precedence. For the action **Update** this will be done first using *property Permissions*, and then *BiskitDef Permissions* and finally the action **Update** *Permissions*. This means all *property Permissions* on the parent will be checked before any *BiskitDef Permissions* on the child will be checked.

© 2010-2015 Exprodo Software

Conditions

The *conditions* assigned to a *Permission* provide a means of very precise control over the situations in which a *Permission* will apply. For example, there may be different *Permissions* that apply depending on:

- the *resource* a *booking* is made for
- the *status* 447 of a *booking* (for example, a *booking* may be editable while it is a request, but not once it has been approved)

or a particular *Permission* is to apply when:

- approving a booking
- cancelling a booking

All of these things are specified by the *conditions*. Here are some examples. Suppose there are two *Permissions*, both of which apply to updating *bookings*, and both of which are *Biskit Type Permissions* (which means they specify the *action* and *Biskit Type* but not a *property*). *Permission* A has the *condition*:

- *status* equals **Requested**

and *Permission* B has the *condition*:

- *resource* equals **Wet Lab**

In this case, assuming both *Permissions* apply to the current user, then *Permission* A will apply whenever the user tries to update a *requested booking* 449 and *Permission* B will apply whenever the user tries to update a *booking* for the **Wet Lab** resource.

What happens when trying to update a *requested booking* for the **Wet Lab**. In that case, both *Permissions* could apply. This is where the *Permission* priority comes in. If one *Permission* has a higher priority than the other, then the highest priority *Permission* will be used. If they have the same priority, then either one could be used.

Authorising And Non-authorising Permissions

When thinking about giving *permission* for something, one normally thinks in positive terms. That is, authorising *permission* for something. However, the layering used in **Calpendo** requires that there is also the notion of non-authorising *permissions* as well as authorising ones. That means that a *Permission* can be written that will grant the right to do something at one layer, and then have another *Permission* at another layer that will explicitly deny the right to do something.

For example, there may be an authorising action *Permission* that allows everybody to read everything, and then a non-authorising *property Permission* to deny the ability to read the *booking* price for some people.

© 2010-2015 Exprodo Software

Targeting Users

For each *Permission*, choose which users it will affect.

However, it's sometimes easier to say who a *Permission* shouldn't affect, or to say that it's everybody meeting some criteria, apart from those that meet some other criteria.

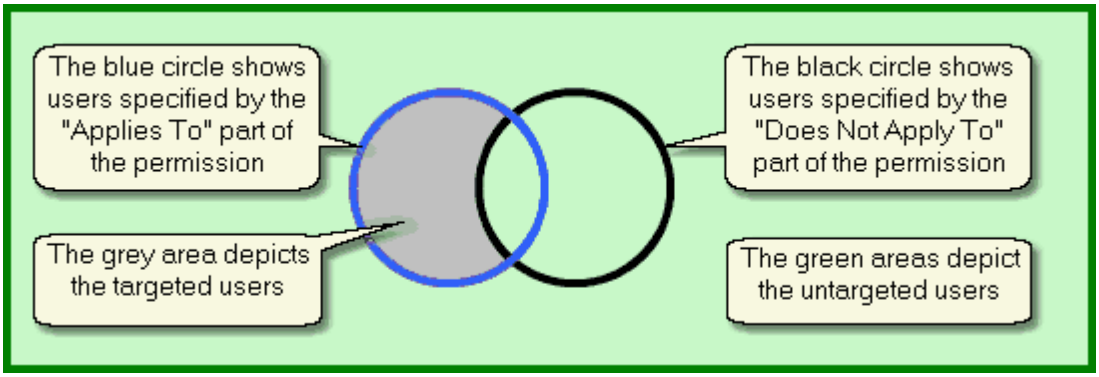
So **Exprodo** applications allows the specification of the users a *Permission* *applies to*[446] as well as those it *does not apply to*[447]. The affected users will be everybody the *Permission* applies to that are not amongst those it *does not apply to*.

Within each of the *applies to* and *does not apply to* sections, identify users by employing one or more of the following methods:

- Name individual users or *user groups*[450].
- Specify users by the *roles*[450] they have.
- Specify a property on the object being controlled that will identify a user or *user group.*

This last item, specifying a *property* that identifies a user or *user group*, needs more explanation. For example, a Calpendo *booking* has an *owner*[448] that is the user that owns the *booking*, a *booker*[446] that is the user that created the *booking* and a *project*. Each *project* also has an *owner* and users.

Consequently, when writing a *Permission* that affects a *booking*, users can be targeted by using any of the *properties owner*, *booker*, **project.owner** and **project.users**.


The blue circle shows users specified by the "Applies To" part of the permission
The black circle shows users specified by the "Does Not Apply To" part of the permission
The grey area depicts the targeted users
The green areas depict the untargeted users

Showing the targeted and untargeted users, defined by the "Applies To" and "Does Not Apply To" users.

© 2010-2015 Exprodo Software

Available Actions

The following table shows the available actions, and indicates whether each action uses *action*, *Biskit Type* or *property Permissions*:

Action	Applies When You...	Uses Action Permissions	Uses Biskit Type Permissions	Uses Property Permissions
Create	Create a new object	X	X	X
Read	View an object	X	X	X
Update	Update an object	X	X	X
Delete	Delete an object	X	X	
Exists	Display a list of objects	X	X	
Email Readable	Include data in an email	X	X	X
Run Report	Run a report	X	X	
Execute	Use the *Booking Rule Validator* 447	X	X	
Dump Database	Take a backup of the whole database	X		
Update Database Schema	Add or remove tables or columns in the database	X		

Create

When a new object is created, action and *Biskit Type Permissions* for the **Create** action are checked. The *property Permission* is checked to see if the *property* should be rendered read or read/write only by comparing against the default value of the *property*.

Read, Exists and Email Readable

When a user tries to view an object, **Exprodo** applications check whether the user has:

- **Exists** *permission* on the object to check whether the user is allowed to know that the object exists
- **Read** *permission* on each *property*, which means that *property* checks are performed on every *property* on the object.

The value of any *property* on which the user does not have **Read** *permission* will be hidden from the user.

When displaying a list of objects, **Exists** *permission* will be checked so that only those objects that the user is allowed to know exist will be seen. This also means that when generating a report that counts the number of objects matching some criteria, then the count will not include objects for which the user does not have **Exists** *permission*.

When sending an email, each object *property* that is to be included in the text is checked for both **Read** *permission* and **Email Readable** *permission*. Also, for every *property* inserted into the email, the object the *property* comes from is checked for **Exists** *permission*.

© 2010-2015 Exprodo Software

Emails may be read by anybody, and so when **Exprodo** applications check *permissions* for what can be included in an email, it checks whether the special user *nobody* 448 has the relevant *permission* as well as the user that performed the action that caused an email to
be sent. Therefore it is necessary to create a *Permission* that gives **Email Readable** *permissions* to user *nobody* for any information that is required to go into an email. See Special Users 167 for more details on user *nobody*.

Update

When a user tries to update an object, **Exprodo** applications will check for **Update** *permission* on the object and also on each *property*. If *permission* is denied for any *property*, then **Update** *permission* for the whole object is denied. This is different from viewing an object, where being denied *permission* to read one *property* does not stop the user from seeing the other *properties*.

When editing an object, if **Exprodo** applications can determine that *Permissions* prevent the user from changing a *property*, then that *property* will be rendered in an unmodifiable form. This will prevent the user from making changes that would only be rejected later.

Delete

When deleting an object, **Delete** *action* and *Biskit Type Permissions* are checked.

Run Report

Run Report *action Permissions* are checked whenever a report is run. This allows the administrator to prevent some types of report being run (for example, to stop some people from running summary or group reports), or the *Biskit Type* being reported on (for example, to restrict reports about system usage or other things).

To control which reports somebody can run, create a **Run Report** *action Permission* for the *Biskit Type* **Report**. Then place conditions on which reports can be run.

Run Report *Permissions* for any *Biskit Type* other than **Report** will be ignored.

Execute

This is checked when trying to run the *Booking Rule Validator* 447 so that only permitted users may test Booking Rules. This is important because an **Advanced Booking Rule** 256 contains user-supplied Java code that will be executed on the server. To cover the potential security issue, users require **Execute** *permission* on the *Booking Rule* 446.

To allow a user to validate *Booking Rules*, there must be a *Permission* for the **Rule** *Biskit Type* that must have no *conditions*. A future version of **Calpendo** will allow the administrator to control which *Booking Rules* a user can validate. Currently, users can either validate all *Booking Rules* or no *Booking Rules*.

Dump Database

Dump Database action *Permissions* are checked when trying to using the **Database Backup** page to generate a copy of all the data in the database.

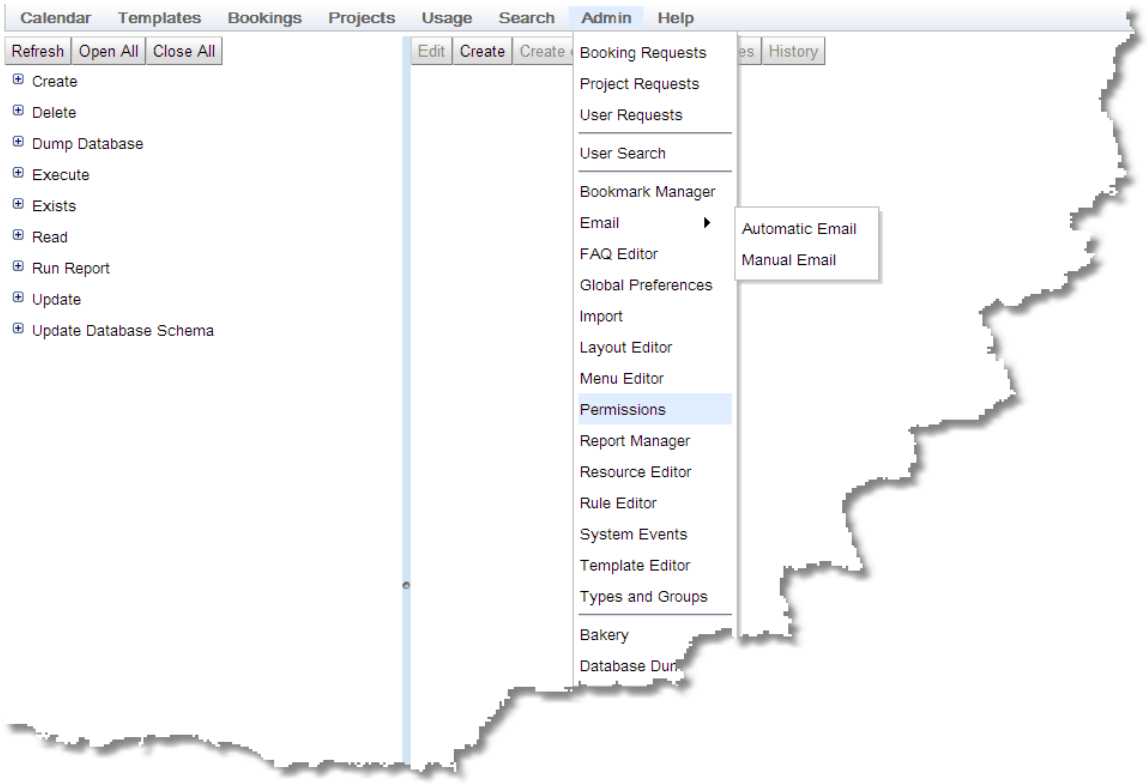
Update Database Schema

Checked when using the *Bakery* 446 to modify the database schema for newly added dynamic *properties*.

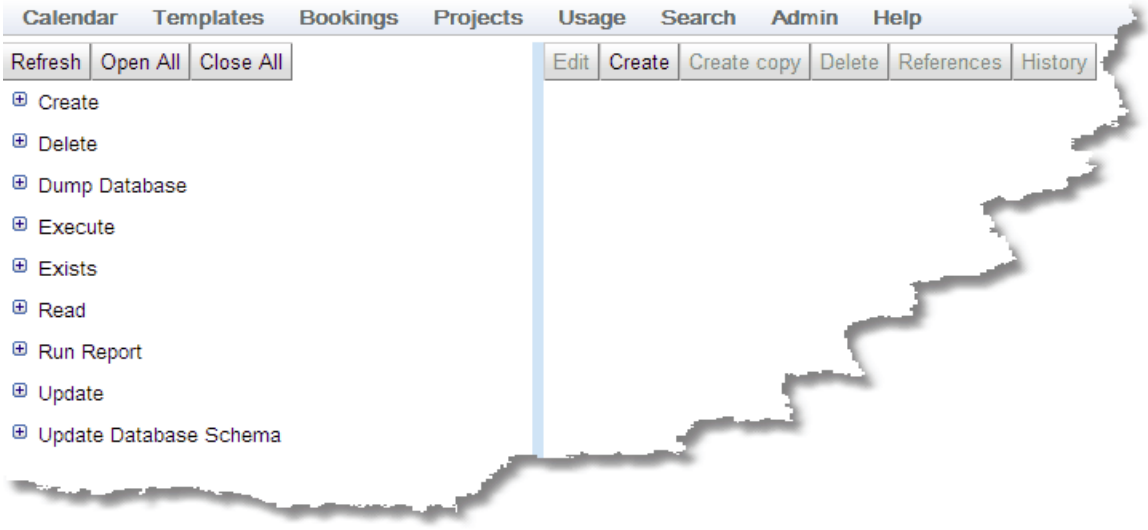
© 2010-2015 Exprodo Software

6.11.2 The Permissions Editor

The **Permissions Editor** shows all the *Permissions* 448 and allows the user to create, update and delete them. By default, it appears on the menu here:



This is what the **Permissions Editor** looks like first opened:



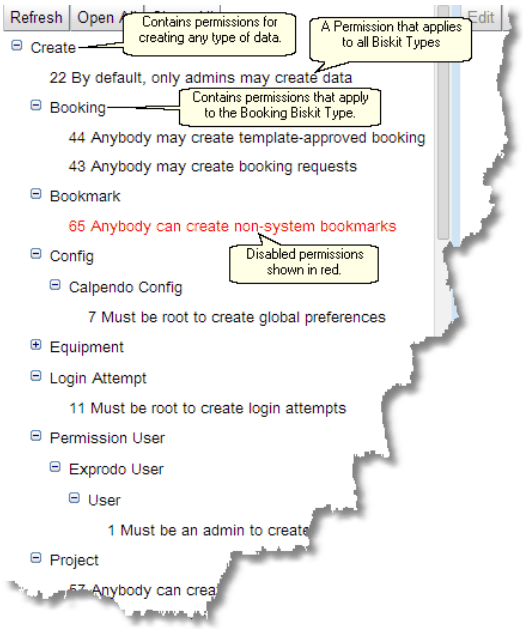
© 2010-2015 Exprodo Software

The Tree Of Permissions

Permissions apply to a particular action[448] and *Biskit Type*[446]. The above screen shot shows all the *Biskit Types* for which there are *Permissions*. Note that a *Permission* can also be defined that will apply to any *Biskit Type*.

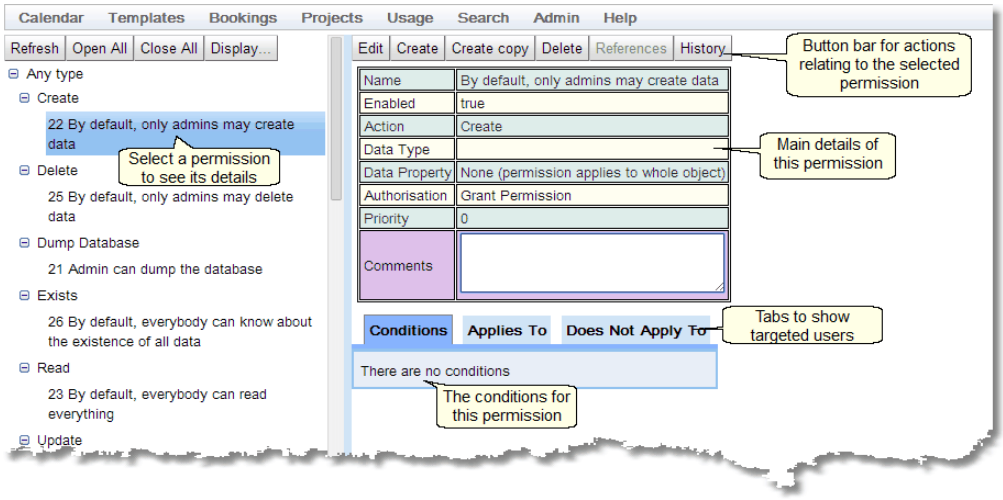
For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition[96] chapter.

Once all the items in the tree are viewable, the *Permissions* are viewable in their *Action*, *Biskit Type*, *Permission* categorisation with any that are disabled shown in red.


Refresh
Edit
Create
Contains permissions for creating any type of data.
A Permission that applies to all Biskit Types
22 By default, only admins may create data
Booking
Contains permissions that apply to the Booking Biskit Type.
44 Anybody may create template-approved booking
43 Anybody may create booking requests
Bookmark
65 Anybody can create non-system bookmarks
Config
Disabled permissions shown in red.
Calpendo Config
7 Must be root to create global preferences
Equipment
Login Attempt
11 Must be root to create login attempts
Permission User
Exprodo User
User
1 Must be an admin to create
Project

Permission Detail

Click on a *Permission* in the tree, and the details for that *Permission* will appear on the right:



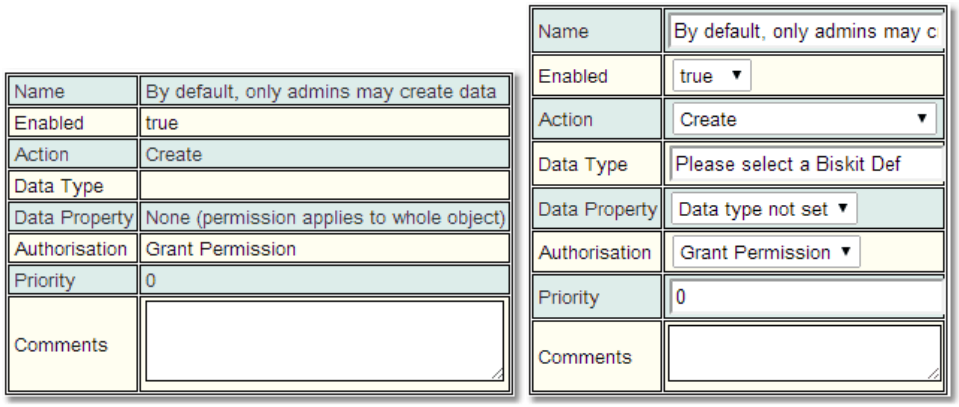
© 2010-2015 Exprodo Software

Now that a *Permission* is selected, the button bar is no longer greyed out:

For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition[96] chapter.

Press the **Edit** button to make the page to be editable, and the button button bar changes:

The main details of the *Permission* contain all the information represented in the tree on the left. These two screen shots show the details in read-only and in edit mode:



When changing these details:

- The **Name** can be as required, but should not be the same as any other *Permission*.
- The **Comments** can also contain anything and is only there to record whatever notes there are about the *Permission*.
- The **Enabled** flag is a way to create a *Permission* without making it take effect immediately, or to turn one off without deleting it. A disabled *Permission* shows in red in the tree.

The other *properties*[448] are described in How Permissions Work[278].

The Conditions Tab

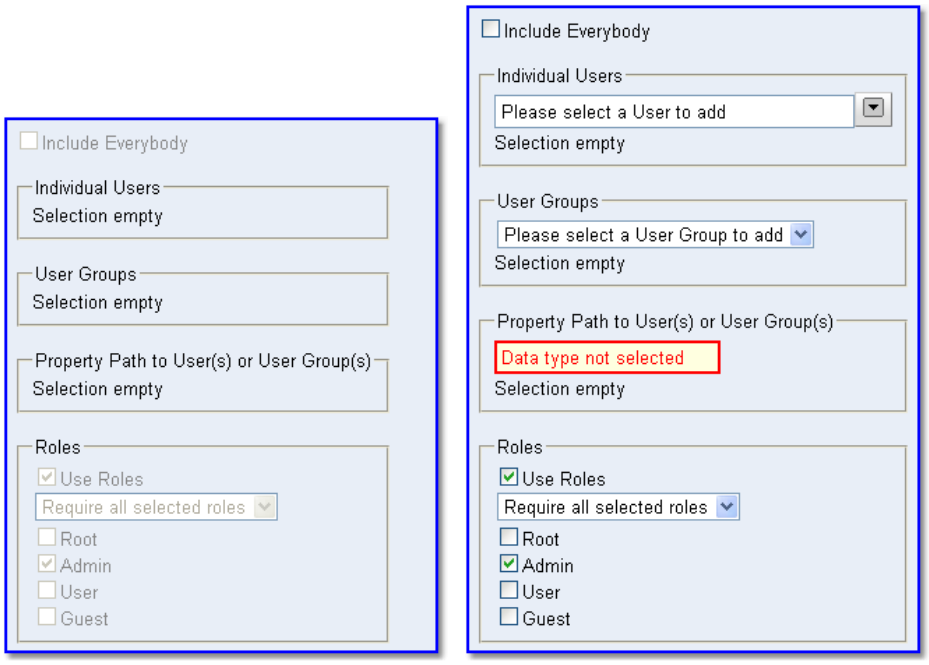
Conditions[447] underpin much of the configuration that can be done in **Calpendo**, and they are used in several places (most notably in *Automatic Emails*[297]). Consequently, they have their own section of this configuration guide. See Conditions[98] for the details.

Please note that the section that describes *conditions* indicates that the **Updated Value** and **Change** *condition* types are considered an error in some contexts. For *Permissions*, they are only usable when the *Permission action* is **Update**.

© 2010-2015 Exprodo Software

The Applies To Tab

In read-only and editable mode, the **Applies To** tab looks like this:



Include Everybody

If set to **true**, then this means that the *Permission* applies to all users, and so the rest of the tab is hidden. As described in Introduction To Permissions [278], the targeted users are those users covered by the **Applies To** tab that are not also covered by the **Does Not Apply To** tab.

Individual Users

This allows the specification of particular users that the *Permission* should apply to.

User Groups

The *Permission* will apply to.any user in the specified *groups* [450].

Property Path

A *Permission* *property path* [449] works exactly the same as the property path [301] in an Automatic Email [297], and it is described more fully in that section. The *property path* is a *property* or list of *properties* that lead from the *Biskit Type* defined on the *Permission* to a user or *user group*, and those users will also have the *Permission* applied to them.

User Roles

Users can have multiple *roles* [450]. Any number of *roles* can be selected by ticking the check box next to the *role*, and then choose with the drop-down whether the *Permission* is to apply to users that have *all* of the selected *roles*, or users that have any of the selected *roles*.

© 2010-2015 Exprodo Software

The Does Not Apply To Tab

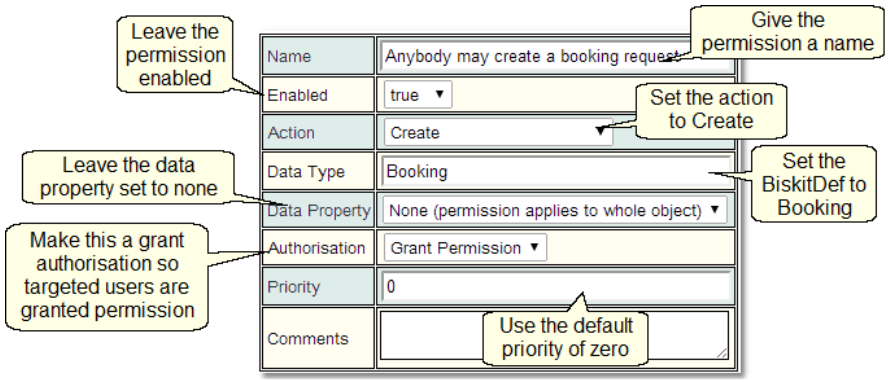
This tab is almost identical to the **Applies To** tab. The only difference is that the check box at the start is labelled **Exclude Nobody** instead of **Include Everybody**. When ticked, it means that no users are taken away from the *Applies To*[446] list of users, and so the rest of the **Does Not Apply To** tab will be hidden. When unticked, the users that should not be targeted are selected.

6.11.3 Example Permissions

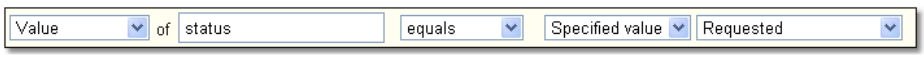
Example 1: Anybody May Create A Booking Request

This example will create a *Permission*[448] that authorises anybody to create a *booking* when the *booking* to be created has its status set to **Requested**. Note that by creating a *Permission* like this, it doesn't influence what happens when somebody tries to create a *booking* whose *status* is set to something other than **Requested**.

1. Create a *Permission* that applies when a *booking* is **Created** and give it a name



2. Add a *condition*[447]: **Status** is **Requested**



3. Under **Applies To**[446], ensure **Include Everybody** is ticked

4. Under **Does Not Apply To**[447], ensure **Exclude Nobody** is ticked

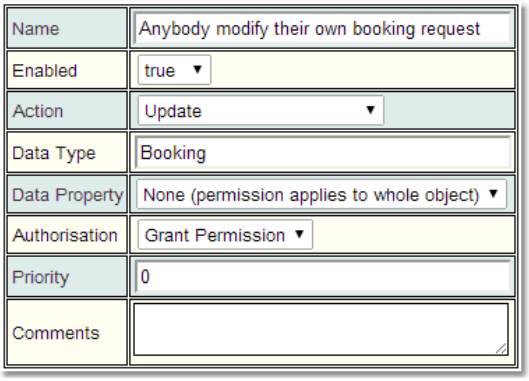
5. Save the *Permission*.

© 2010-2015 Exprodo Software

Example 2: Anybody Can Modify A Booking Request They Created If The Booking Is Still A Request

This example shows the difference between using a **Old Value** condition and an **New Value** condition. We need a *Permission* that will allow somebody to modify a *booking* if they created it and the status was set to **Requested** before they tried to change it, and its value would still be **Requested** after the change.

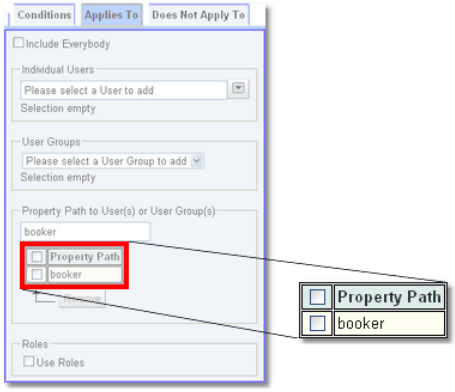
1. Create a *Permission* that applies when a *booking* is **Updated** and give it a name:



2. Add a *condition*: **Status** is **Requested**. Note that for an update, to put a *condition* on the value after the update, then use a **New Value** type of *Condition*.

3. Add a *condition*: **Status** was **Requested**. Here, we use a **Old Value** type of *Condition* to put a *condition* on the value that the status had before the update.

4. We now need to limit the *Permission* to applying to whomever created the *booking*. This is easy because each *booking* stores a record of who booked it in the *booker*[446] *property*. To do this, go to the **Applies To** tab, untick **Include Everybody** and then add a *property* path to *booker*.



5. Save the *Permission*.

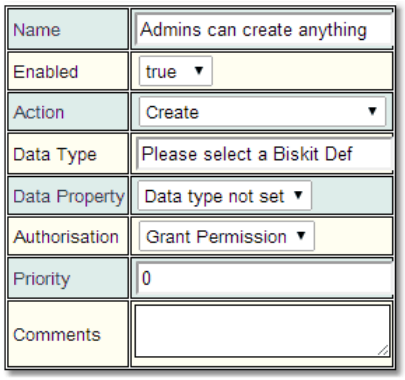
© 2010-2015 Exprodo Software

Example 3: Admins May Create Or Update Anything

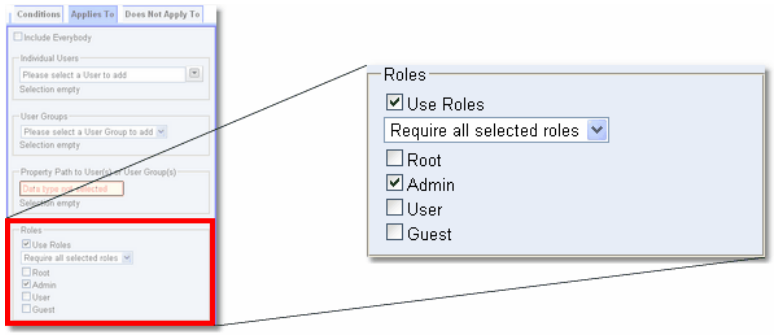
This example shows how to use User Roles to decide who gets *permission* to do something, as well as showing how to create a *Permission* that applies to every *Biskit Type*[446]. Use a *Permission* like this when the Admin Role needs to bestow the ability to create or update whatever they like. The *Permission* can still be revoked in some circumstances by using a *Permission* of a higher level elsewhere that overrides the *Permission* we're creating here.

To achieve the ability to let the administrator create or update anything, we need to create two separate *Permissions*, one for the **Create** and one for the **Update**.

1. Create a *Permission* that applies when **Anything** is **Created** and give it a suitable name. Leave all the other options at their default values:



2. Under **Applies To**, untick **Include Everybody**, select **Use Roles** and tick the check box next to **Admin**



3. Save the *Permission*.

4. Repeat step 1, but this time set the *Action*[448] to **Update** and set the *Permission's* name accordingly.

5. Repeat of steps 2 and 3 above for this **Update Permission**.

© 2010-2015 Exprodo Software

Example 4: Only Allow Specified Users To See The Price Being Charged For Use Of A Resource

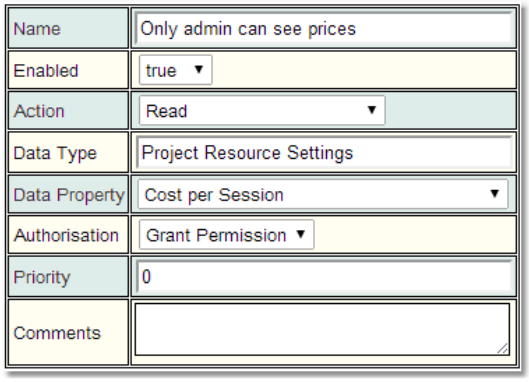
This example protects the **Cost Per Session** property of a *Project Resource Setting*, as stored in each *project*. The idea is that we would like to stop anybody apart from one particular administrator being able to see what the price being charged is. This serves to demonstrate how to protect individual *properties* and how to apply Permissions to individuals. This is known as a Property Level Permission 279.

There are two ways we can set up *Permissions* to achieve the result we want, depending on what we'd like to achieve. We can create a *Permission* for each methodology:

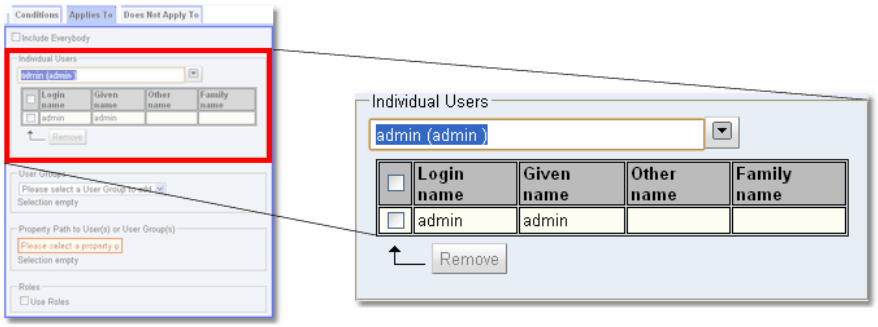
- Authorise the user **Admin** to read the the **Cost Per Session**, but say nothing about whether other users can read it.
- Deny everyone other than the user **Admin** from being able to read the **Cost Per Session**, but say nothing about whether **Admin** can read it.

When a *Permission* says nothing about whether a user is granted or denied *permission*, then you rely on other *Permissions* to specify what you want. This is the way that multiple *Permissions are* superimpose to get the result required. To show how this works, we'll go through each of the options above in turn.

1. Create a *Permission* that applies when a *Project Resource Setting* is **Read** and set the **Data Property** to **Cost Per Session**



2. On the **Applies To** tab, untick **Include Everybody** and under **Individual Users**, select one or more users. In this example, a user called **Admin** has been added.



© 2010-2015 Exprodo Software

3. Save the *Permission* as it is, and with the default values of **Authorisation** being **Grant Permission,** it means that the targeted users (in this case, the user **Admin**) will be granted *permission* when trying to view the **Cost Per Session**. This *Permission* says nothing about whether other users can read **Cost Per Session**.

If you now wanted to change the *Permission* so that it says nothing about whether **Admin** can read **Cost Per Session**, but that it denied *permission* for all other users, follow the following steps:

4. Press the **Edit** button (unless you didn't actually save it and so it's still shown in an editable form).

5. On the **AppliesTo** tab, tick **Include Everybody**.

6. On the **Doesn't Apply To tab**, untick **Exclude Nobody** and add **Admin** to the list of individual users.

7. In the main *Permission* details, change **Authorisation** to **Deny Permission**.

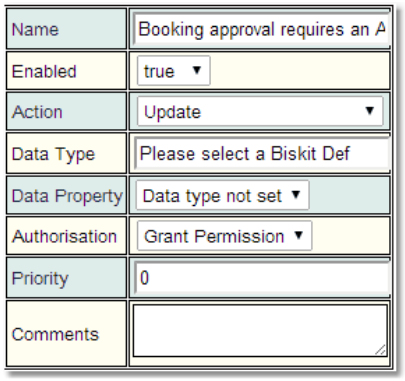
What we've now done is to target everybody apart from **Admin**. We've made the *Permission* negative, so that the targeted users will be denied permission to read **Cost Per Session**.

© 2010-2015 Exprodo Software

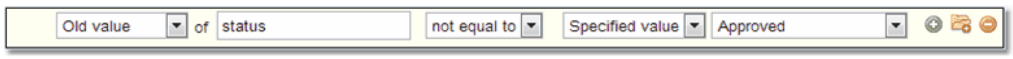
Example 5: Approval Of A Booking Requires An Admin

In this example, we are going to create a *Permission* that allows anybody with the Admin Role to change a *booking's status* 447 to **Approved.** We will also add a *condition* to make sure we're matching the situation where an administrator modifies the *status* themselves, rather than where the *status* is modified indirectly as a result of moving the *booking* to a period in which a *Time Template* 450 has approved the *booking*.

1. Create a *Permission* that applies when a *booking* is **Updated**, give it a name, and make it **Grant Permission**.



2. Add a *condition*: **Old value Status** was not **Approved**



3. Add a *condition*: **New value Status** is **Approved**

4. Add a *condition*: **New value templateApproved** is **false**. The **templateApproved** *property* is set automatically when a *booking* is approved by a *Time Template* and so we can use it to detect how the status was changed.

5. Under **Applies To**, untick **Include Everybody**, select **Use Roles** and then add **Admin**.

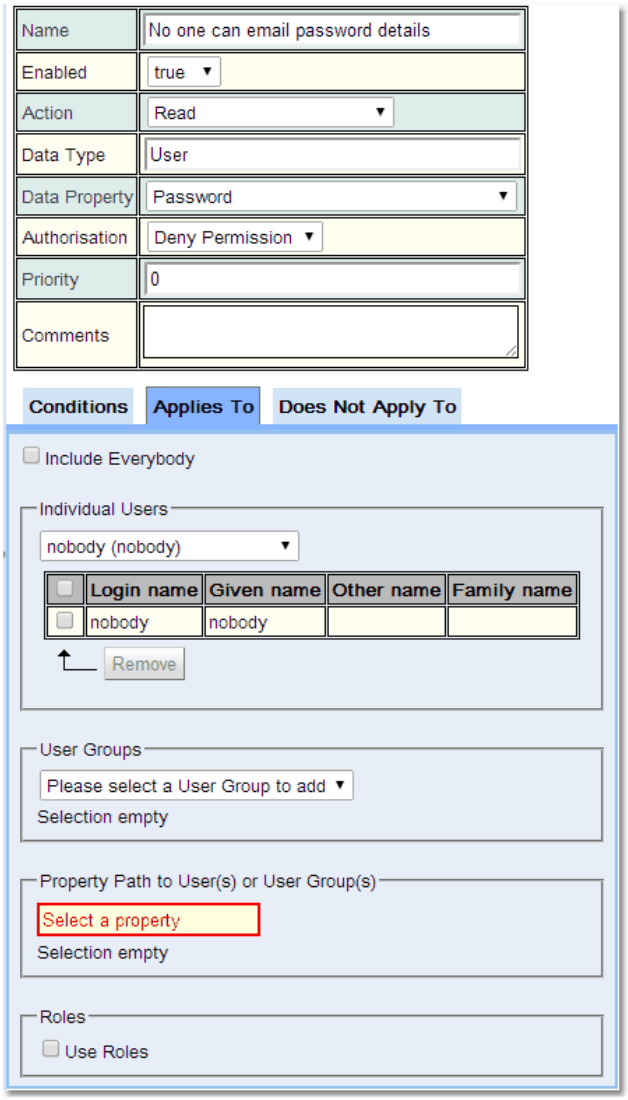
6. Save the *Permission*.

© 2010-2015 Exprodo Software

Example 6: No One Can Email Password Details

In this example, we are going to create a *Permission* that stops everybody from putting the value of the password property of the **User** *Biskit Type* into an email. To do this we will be using the special user *nobody*[448]

1. Create a *Permission* that applies when a **User** is **Read**, give it a name, select the *property* **Password**, make it negative (**Authorisation** is **Deny Permission**).



2. Under **Applies To**, untick **Include Everybody**, select **Individual Users** and then *nobody*.

3. Save the *Permission*.

Use this mechanism on any information you do not want sent by e-mail.

© 2010-2015 Exprodo Software

Example 7: Hiding Resources And Booking Properties

Sometimes the administrator may want to create *resources* that only some people may access, or define *properties* on a *booking* that some people shouldn't be allowed to see. This is achieved by setting up *permissions* that control who can access what.

To hide some *bookings* from some people:

- Deny them READ *permission* on the *bookings* they should not be able to see.

If you want to hide a *booking property* from some people, then:

- Create a property-level Booking *permission* that denies them READ rights on that *property*
- Create a separate *permission* for each *property* you need to hide

If you want to hide a *resource* from someone, here is the list of *Permissions* that need to be set up:

- Deny them EXISTS *permission* on the *resource*
- Deny them READ *permission* on *bookings* for that *resource*
- Deny them READ *permission* on *templates* for that *resource*.

To set up the first permission.

Action	Exists
Data Type	Resource
Condition	Value of self equals Specified value (name of resource)

Self is *property* of type *Biskit* that points to the *Biskit* owning the *property*. By using the **self** *property* rather than the **name** *property* the primary key of the *Biskit* is stored in the *Condition* and not the name, therefore if the name changes the *Permission* will still work.

© 2010-2015 Exprodo Software

6.12 Automatic Emails

Calpendo can send emails automatically, in response to any data being created, updated or deleted. An **Automatic Email** is a specification for the conditions that should generate emails, to whom they should be sent, and what the content of those emails should be. So every time any data is created, updated or deleted, **Calpendo** checks to see if there is an **Automatic Email** that matches the data that was created, updated or deleted. The **Automatic Email** doesn't just specify fixed content that should be sent in the email, but rather, it specifies a template so that *properties* [448] from the changed data can be included.

When creating an **Automatic Email**, these are the elements set:

- A **Name**, to can distinguish it from other **Automatic Emails**.
- The *Biskit Type* [446] of the data that must change for the **Automatic Email** to trigger.
- The type of change the data must undergo. That is, **Create**, **Update** or **Delete**.
- Optionally, specify *conditions* [447] that are tested against the changed data.
- Specify the recipients of the email. Do so by specifying:
    1. Individual Users
        a. For occasions when the *Biskit Type* being changed is a **User**, *User Type* [450] or *User Group* [450], then that user (or set of users) may be required to receive an email. For example, if an administrator changes a user's status to **Blocked**, then set up an **Automatic Email** that sends the blocked user an email telling them they've been blocked.
        b. The user that caused the original data change should receive the email. For example, when a user creates a *project* [448], send them an email to confirm its creation.
    2. Individual *user types*
    3. Individual *user groups*
    4. The *roles* [450] that a user should have, for example to send the email to all users with the **Admin** *role*.
    5. A path to a custom *property* storing users, *user types*, *user groups* or email addresses. This gives the ability to send emails to users outside the **Calpendo**.
- Specify the subject of the email. This can contain *properties* from the changed data.
- Specify the content of the email. This can contain *properties* from the changed data.

There is a flag to be set on the ***A*utomatic Email** to choose whether it is enabled or not. This allows the creation of **Automatic Emails** that are not always in use, and therefore can be turned on or off as required.

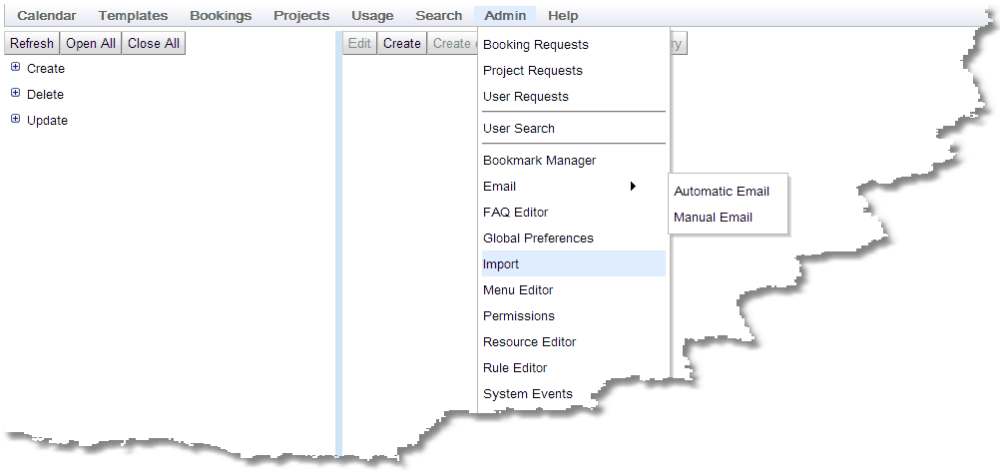
See Also:

Conditions [98] - these are central to the way **Calpendo** works and **Automatic Emails** in particular.

© 2010-2015 Exprodo Software

6.12.1 The Automatic Email Editor

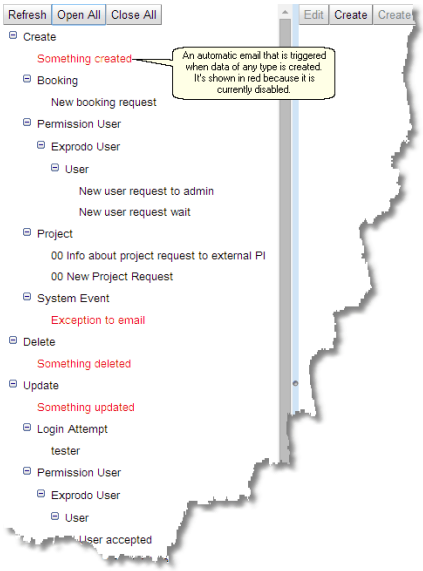
The **Automatic Email Editor** shows all the **Automatic Emails** and allows creating, updating and deleting them. By default, it appears on the menu here:



The Tree Of Automatic Emails

Automatic Emails are triggered when data of a particular *Biskit Type* [446] is created, updated or deleted. Each **Automatic Email** defines which *Biskit Type* triggers it and which action (created, updated or deleted) triggers it. This means all the **Automatic Emails** can be categorised by *Biskit Type* and action. The above screen shot shows all the *Biskit Types* for which there are **Automatic Emails**. Note that an **Automatic Email** can be defined that will trigger for any *Biskit Type*.

For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition [96] chapter.



Once all the items in the tree are viewable, the emails in their *Biskit Type*, **Action**, **Automatic Email** categorisation will be seen, with any that are disabled shown in red. The tree shown here shows fourteen **Automatic Emails**, four of which are disabled and so are shown in red.

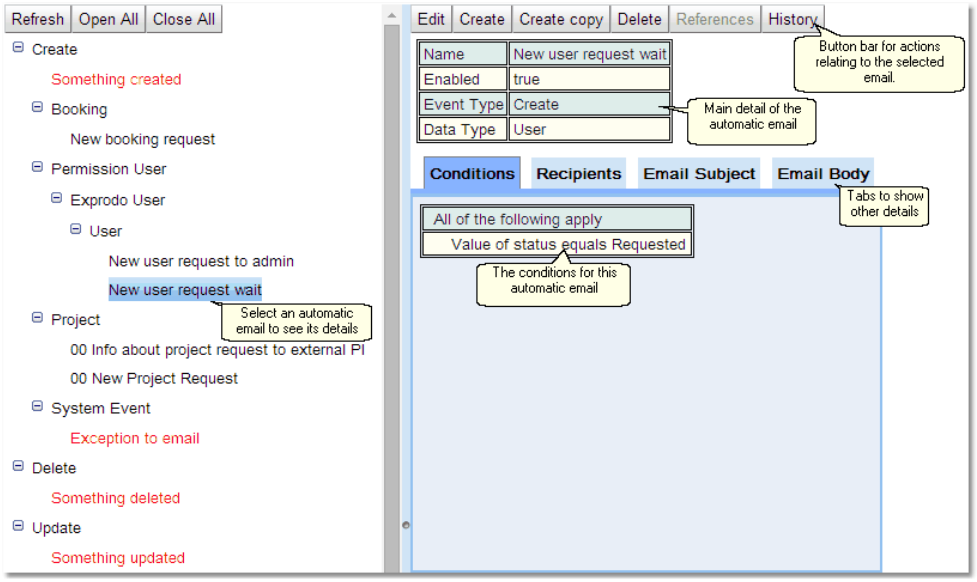
The first one shown under User, called **New user request to admin**, is an **Automatic Email** that is triggered when a user is created.

This doesn't mean that an email will actually be sent, because the **Automatic Email** may have *conditions* [447] that need to be checked. In this case, it has a *condition* that checks whether the *user's status* [450] is **Requested**, although you can't tell that without clicking on the **Automatic Email** to see its details.

© 2010-2015 Exprodo Software

Automatic Email Details

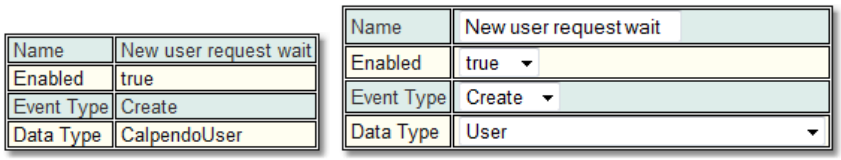
Click on an **Automatic Email** in the tree, and see its details appear on the right:



For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition[96] chapter.

Press the **Edit** button to make the page editable, and the button bar changes:

The main details of the **Automatic Email** contain all the information represented in the tree on the left. These two screen shots show the details in read-only and in edit mode:



When changing these details:

- The **Name** can be anything you like, but must be unique amongst **Automatic Emails**.
- The **Enabled** flag is a way to create an **Automatic Email** without making it take effect immediately, or to turn one off without deleting it. These show in red in the tree.
- The **Event Type** is where it is specified whether the **Automatic Email** is triggered by a ***Create***, **Update** or **Delete**.
- The *Data Type* value indicates which *Biskit Type* triggers the email. Choose the **Any data type** option to make the **Automatic Email** trigger for every *Biskit Type*. However, this does quite severely limit what Conditions[98] you can use and what "magic text" can be inserted into the email (described below).

© 2010-2015 Exprodo Software

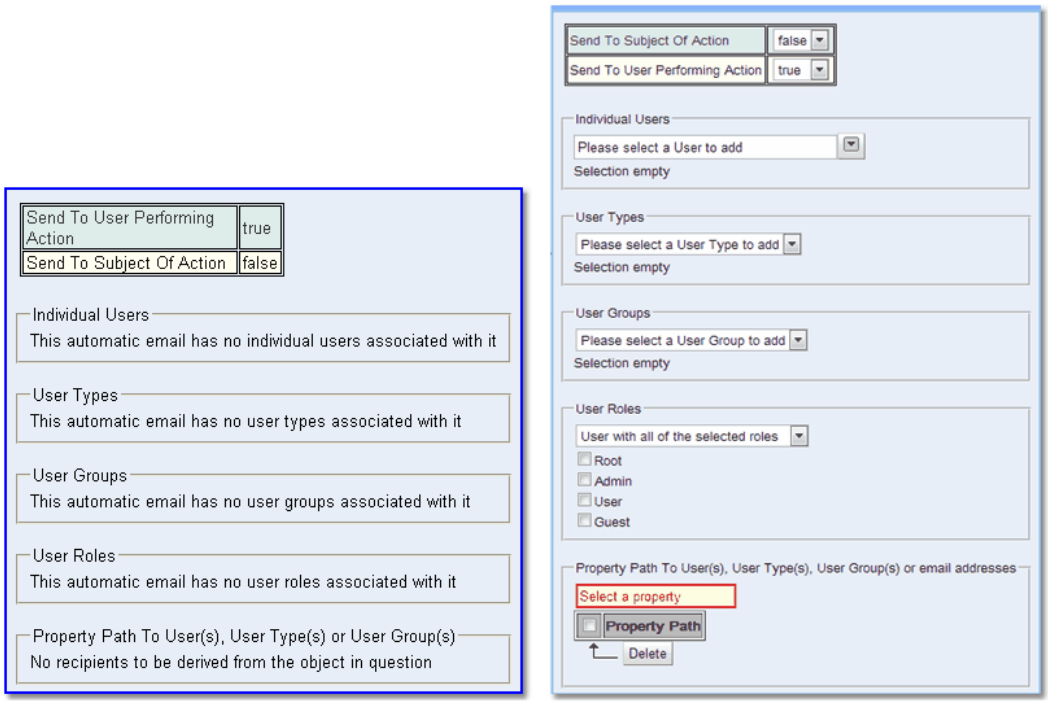
The Conditions Tab

Conditions underpin much of the configuration that can be done in **Calpendo**, and they are used in several places (most notably in **Automatic Emails** and Permissions)[278]. Consequently, they have their own section of this configuration guide. See Conditions[98] for the details.

Please note that the section that describes *conditions* indicates that the **Updated Value** *condition* type is considered an error in some contexts. When the **Automatic Email** is triggered by a **Create** or **Delete** action, that is an example of **Updated Value** being inappropriate.

The Recipients Tab

In read-only and editable mode, the **Recipients** tab looks like this:


Send To User Performing Action
true
Send To Subject Of Action
false
Individual Users
This automatic email has no individual users associated with it
User Types
This automatic email has no user types associated with it
User Groups
This automatic email has no user groups associated with it
User Roles
This automatic email has no user roles associated with it
Property Path To User(s), User Type(s) or User Group(s)
No recipients to be derived from the object in question
Send To Subject Of Action
false
Send To User Performing Action
true
Individual Users
Please select a User to add
Selection empty
User Types
Please select a User Type to add
Selection empty
User Groups
Please select a User Group to add
Selection empty
User Roles
User with all of the selected roles
Root
Admin
User
Guest
Property Path To User(s), User Type(s), User Group(s) or email addresses
Select a property
Property Path
Delete

Send To Subject Of Action

This option is only applicable when the data whose change triggered the Automatic Email is itself a **User**, *User Group*[450] or *User Type*[450]. When setting this option to **true**, and triggered by a change to a **User**, then the changed user will receive an email. This means that when a **User** is modified (for example to change their *status*[450]), they can receive an **Automatic Email**. Similarly, if a *User Group* or *User Type* changes, then send an email to everybody in that *User Group* or with that *User Type*.

Send To User Performing Action

An **Automatic Email** is triggered when some data is changed. Normally (but not always), that change is triggered by something a particular user did. If that user is required to receive the email, then select **true** for **Send To User Performing Action**.

© 2010-2015 Exprodo Software

Individual Users

This lets you specify particular users that should receive the email.

User Types

If you specify one or more *User Types*, then any user with one of those *User Types* will receive the email.

User Groups

Any user in the specified *groups* will receive the email.

User Roles

Users can have multiple *roles*[450]. Select any number of *roles* by ticking the check box next to the *role*, and then choose with the drop-down whether the email is to go to users that have all of the selected *roles*, or users that have any of the selected *roles*.

Property Path

As an abstract concept, the *property path*[449] may at first appear confusing. It is used when the data that is changed contains a reference to a **User**, *User Group* or *User Type*, or else contains a *reference*[449] to something that contains a *reference* to a **User**, *User Group* or *User Type*. For example, a *System Event*[449] is a record of various things that happen. Every time a user logs in, a *System Event* is created. *System Event* has a *property*[448] called user that records who caused the event. If you were to create an **Automatic Email** that was triggered by a *System Event*, then direct the email to the user that caused that event by choosing a *property path* of **user**.

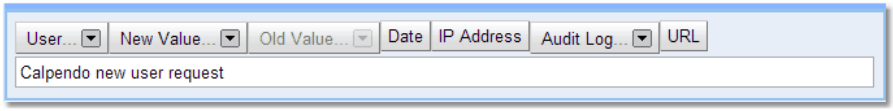
Custom *properties* can also be used which hold email addresses, this allows **Calpendo** to send emails to users outside **Calpendo**, by choosing such a *property* the **Automatic Email** would extract the email address.

Once the *Biskit Type* is selected that the **Automatic Email** is triggered by, then clicking in the *Property Path* selector will show a drop-down panel with all the possible paths that lead to a **User,** *User Group, UserType* or **Email property**.

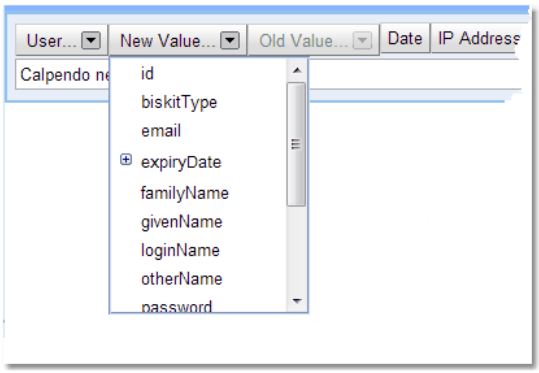
© 2010-2015 Exprodo Software

The Email Subject Tab

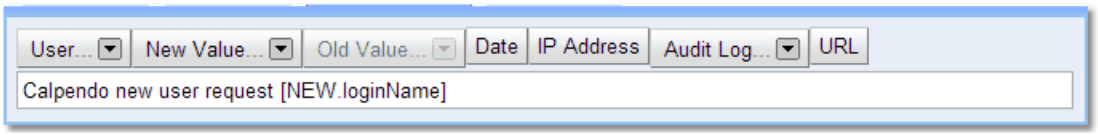
In write mode, the email **Subject** tab looks like this:



In the case shown, the email would have a subject of **New Calpendo user request**. The buttons that appear in the tab enable dynamic text to be inserted that will be replaced when the email is generated. For example, suppose the email subject is required to be **New Calpendo user request for X** where X should be replaced by the login name of the new user. To do this, first append the text "for " to the subject, making sure to leave the cursor at the point where the dynamic text is required to appear. Then press the **New Value...** button and select the user's login name from the drop-down:



And this would insert text into subject so that it looks like this:



The text inserted could have been typed in manually if it was known what to type. The buttons provide easy ways to enter the right **magic** text. When the **Automatic Email** is saved, the email **Subject** tab changes its appearance to look like this:

Once saved, the **magic** text parts are parsed and rendered differently, in this case in blue and bold. This gives some visual feedback so that it can be seen that the text has been recognised correctly. This is particularly useful if **magic** text has been manually typed in, rather than using the buttons.

© 2010-2015 Exprodo Software

The User... Button

The **User...** button inserts text that is replaced by the user that performed the action that led to the **Automatic Email** being triggered. If an administrator were to create a new **User**, then the **User...** button would insert details about the administrator and not the newly created **User**. If somebody registers themselves with **Calpendo**, then the **User...** button would insert details about the new user, since they instigated the event that generated the email.

The New Value... And Old Value... Buttons

If a *Biskit Type* of **Any data type** has been selected for this Automatic Email, then these buttons cannot be used. When a specific *Biskit Type* is selected, then these buttons generate a pop-up with *properties* that belong to that *Biskit Type*.

This means that they allow the insertion of text from the object that was created, updated or deleted. In the case of an update triggering the email, then the **New Value...** button is linked to the *properties* on the object after the change has taken place and the **Old Value...** button is linked to the *properties* on the object before the change took place.

For date *properties* extract just the date or just the time to give more control over the content of the subject.

The example above is a special case: since the **Automatic Email** is triggered by a **User** changing, then this means that the **New Value...** and **Old Value...** buttons show the *properties* of a ***User***, just like the **User...** button.

When the action triggering the email is either **Create** or **Delete**, then the text inserted by the **New Value...** and ***Old Value...*** buttons will generate identical results.

The Date Button

This inserts the text **[DATE]** and when the email is generated, this inserts the date and time that the email is generated.

The IP Address Button

This inserts the text **[IP ADDRESS]** and generates the network IP address of the user that generated the change. For changes initiated by the **Calpendo** server, rather than a particular person, this will show the server's loopback network address.

The AuditLog... Button

Whenever data is created, updated or deleted, there is an **AuditLog** entry created. This button allows the extraction of information from the audit log entry. This is not normally useful, but is required for a special case. When a user registers, there is normally an **Automatic Email** that lets an administrator know that a user has just registered. That email contains two **URLs** - one that will automatically approve the user, and one that will automatically deny the user. For this to work, **Calpendo** needs to make sure that when the relevant **URLs** are accessed, nobody else has already changed the user's *properties*. So, the **Automatic Email** uses the audit log information to add the ***AuditLog's*** identifier. This is a unique identifier for the audit log entry. **Calpendo** can then examine the audit log and check to see if there have been any changes to the user since the time the email was generated; only if there have been no subsequent changes will the user be approved or denied.

© 2010-2015 Exprodo Software

The *URL* Button

This inserts a **URL** to your **Calpendo**. Sometimes, it is required to provide a link in an email to a particular page. This can be done by putting the relevant page information after the main **URL**. For example, if a *Project*[448] has been changed, then set up a link like this: **[URL]/#ba&type=Project&action=view&id=[NEW.id]**

In this case, **[URL]** is replaced by the main **URL**, then **#ba&type=Project&action=view** links to a page that is told to display *Projects* and not other *Biskit types*. Finally, the **&id=[NEW.id]** specifies the identifier of the particular *Project* that was changed.

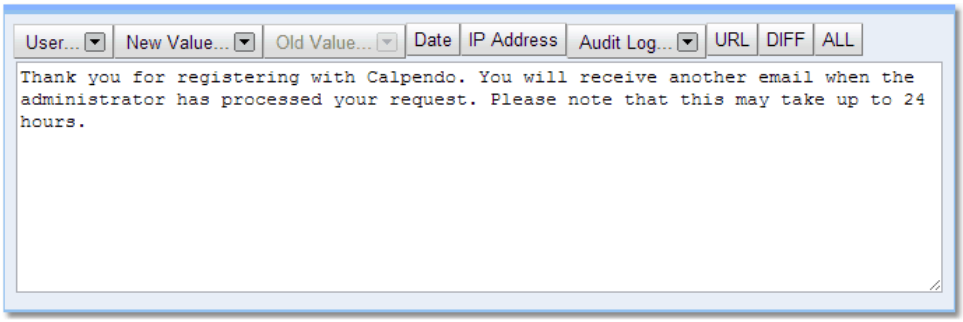
The easy way to work out how any URL like this should be formatted is to first go to the relevant page and see what URL **Calpendo** uses, then replicate that within the **Automatic Email**, replacing parts as appropriate with dynamic text.

It should be noted that it's not normally useful to insert a **URL** into an email's subject. However, the same **URL...** button is also used for the content of an email (as discussed in the next section).

© 2010-2015 Exprodo Software

The Email Body Tab

The body of the email works just like the subject does. The only differences are that there are extra buttons, for additional **magic** text, and the editor allows multi-line text instead of single-line text. In write mode, it looks like this:


User...
New Value...
Old Value...
Date
IP Address
Audit Log...
URL
DIFF
ALL
Thank you for registering with Calpendo. You will receive another email when the administrator has processed your request. Please note that this may take up to 24 hours.

The extra two buttons, **DIFF** and **ALL** generate a large amount of data. When triggered by an update, **DIFF** inserts text that is converted to something that shows what changed. **ALL** generates output that shows every piece of data from the changed *Biskit*[446], both before and after the change. This can be particularly noisy. The information it shows is normally everything that can be reached from the **New Value...** and **Old Value..** buttons. The exception to this is that if the **Automatic Email** is triggered by any *Biskit Type*, then the **New Value...** and **Old Value..** buttons cannot display appropriate *properties* because it doesn't know what *Biskit Type* will trigger the email. You also cannot type in something like **[NEW.name]** and hope that it will work, because it will not. In those cases, **[ALL]** is the only way to get at the details. The format this generates is not particularly user-friendly, and so if you use this, you should be careful about who receives such emails.

In order for any data to be included in an email it needs to have **Read** and **Email Readable** *Permissions*[448] set for the special user *nobody*[448]. By default there are no **Email Readable** *Permissions* set up, so no data can be sent by email. In order to allow data to be sent by email go to Permissions[278] and create an **Email Readeable** *Permission* for the user *nobody*.

Dealing With Sensitive Data In Emails

If there is data that any user shouldn't be able to put into an email because of confidentiality or sensitivity but the user still needs to be able to access the information in **Calpendo** then set up specific *Permissions* for data. There is a special user called *nobody* which is used for this purpose. In the **Permissions** page, for a *property* or *Biskit,* create a *Permission* with *Action* set to **Read, E-Mail**, **Readable** or **Exists**, **Positive** to **False** and *Applies To*[446] to *nobody*, so that the *property* or *Biskit* can still be accessed and read by the user within **Calpendo** but cannot be added to an email. Read the Permissions[278] section for more details on setting *Permissions*.

© 2010-2015 Exprodo Software

6.12.2 Example Automatic Emails

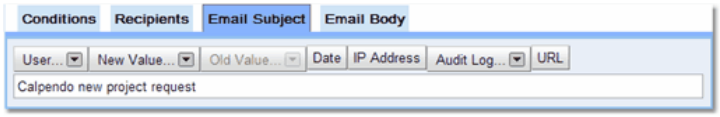
Example 1: Send Email To An Administrator When A Project Request Created

This first example shows how to send an email to an administrator when somebody creates a *Project*[448] whose *status*[448] is **Requested**.

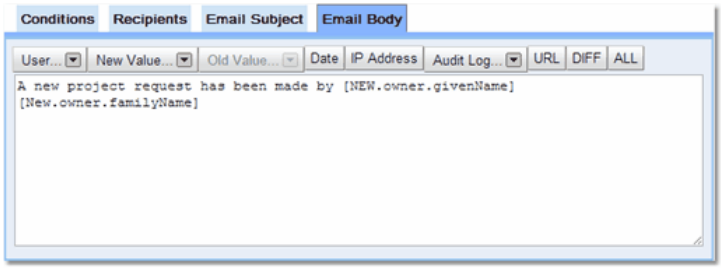
1. First, go to the **Automatic Emails** page and press the **Create** button to create a new Automatic Email.
2. Give it a name, like **Project request to admin**
3. Specify the *Biskit Type*[446] as **Project**
4. Specify an action type of **Create**

5. Add a *condition*[447]: **Status** is **Requested**

6. Add the recipients you require
7. Enter the email's subject



8. Specify the text of the email. This can reference *properties* of the project created and the user who created it.

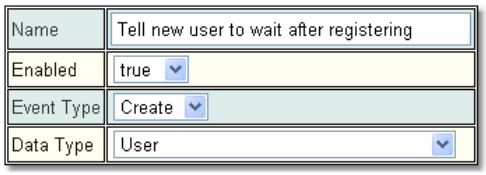


© 2010-2015 Exprodo Software

Example 2: Send Email To A New User After They Register

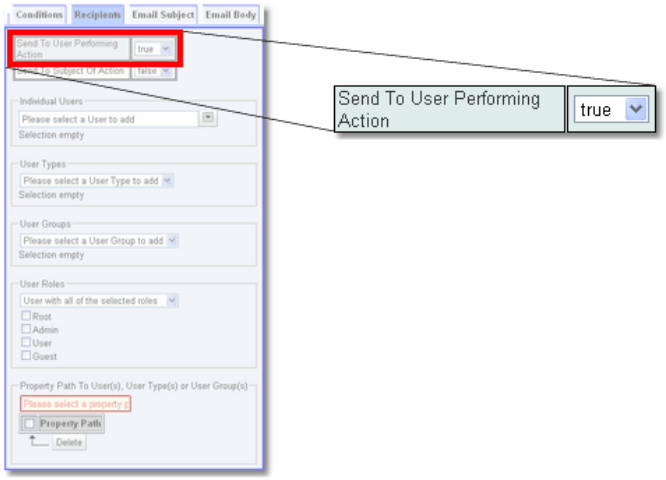
This example shows how to create an **Automatic Email** that will be sent to a new user immediately after they register with **Calpendo**. We need to make sure it gets sent when a new **User** is created, and that the email gets sent to the new user.

1. Create an Automatic Email that is triggered when a **User** is **Created**.

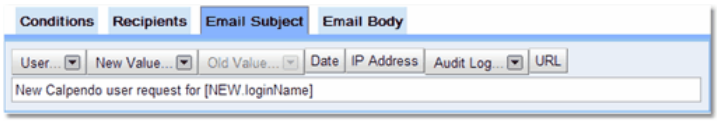


2. Add a *condition*: **Status** is **Requested**

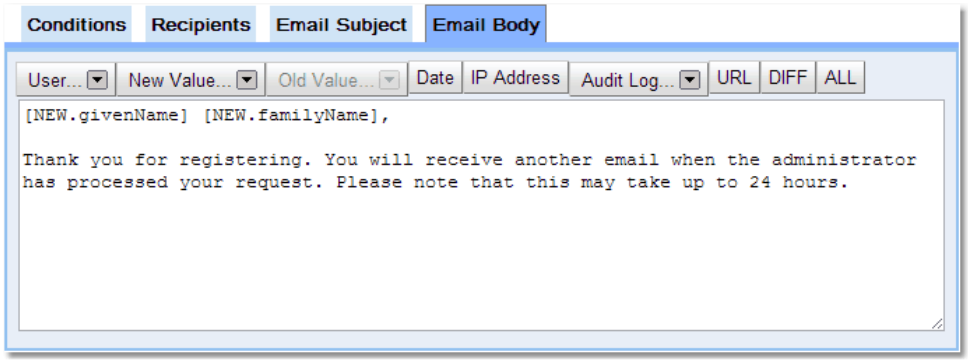
3. On the **Recipients** tab, set **Send To User Performing Action** to **true**.



4. On the **Email Subject** tab, enter the text you want to be the subject of the email.



5. On the **Email Body** tab, enter the text you want to be the main content of the email.

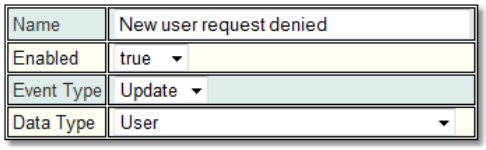


© 2010-2015 Exprodo Software

Example 3: Send Email To A User When New User Request Denied

This example shows how to send an email to user who has just registered (so the **status** is set to **Requested**) but had their request denied (so the **status** has changed to **Denied**).

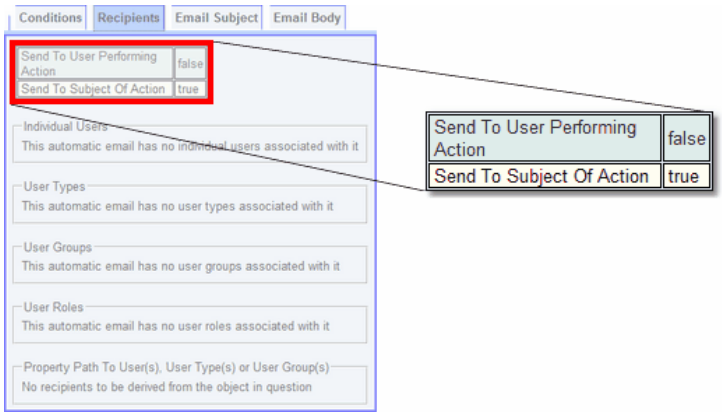
1. Create an Automatic Email that is triggered when a **User** is **Updated**.



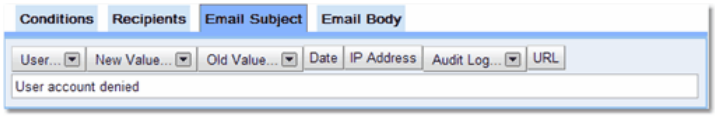
2. Add a *condition*: **Old Value** is **Requested**

3. Add a *condition*: **New Value** is **Denied**

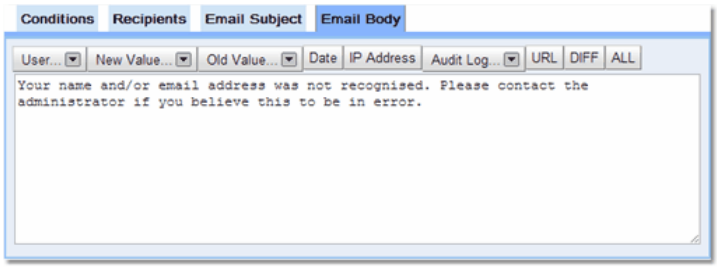
4. Set **Send To Subject Of Action** to ***true***;


Conditions
Recipients
Email Subject
Email Body
Send To User Performing Action
false
Send To Subject Of Action
true
Individual Users
This automatic email has no individual users associated with it
User Types
This automatic email has no user types associated with it
User Groups
This automatic email has no user groups associated with it
User Roles
This automatic email has no user roles associated with it
Property Path To User(s), User Type(s) or User Group(s)
No recipients to be derived from the object in question
Send To User Performing Action
false
Send To Subject Of Action
true

5. Enter the email's **Subject**



6. Enter the email's main content



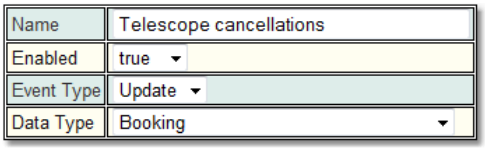
© 2010-2015 Exprodo Software

Example 4: Send An Email When A Booking Is Cancelled

This example shows how you can do quite complicated things by adding *conditions* to achieve the desired result. We're going to create an **Automatic Email** that will tell interested people when there's a cancellation on a particular *resource*[449]. The purpose of this email is to enable *resources* to be better utilised by filling in gaps in the *bookings*[446]. However, if the *booking* being cancelled is too far into the future, there's little point telling people about it, so we will make sure the *booking* is no more than 14 days into the future. Also, if a *booking* has only just been made, then sending out an email is likely to be viewed as spam. So we will make sure the *booking* was created before 15 minutes ago.

For this last requirement to work, we rely on the fact that **Calpendo** stores the date and time a *booking* was created on the **Booking**.

1. Create an Automatic Email that is triggered when a *booking* is **Updated**



2. Add a *condition*: **Resource** is **Clinical Physiology Lab**

3. Add a condition: **Old Value** not equal to **Cancelled**

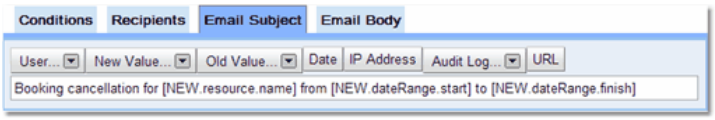
4. Add a *condition*: **New Value** equals **Cancelled**

5. Add a *condition*: **New Value** of **dateRange.start** earlier than **now plus 14 days to the day**

6. Add a *condition*: **Old Value** of **created** earlier than **now minus 15 minutes with accuracy of a minute**

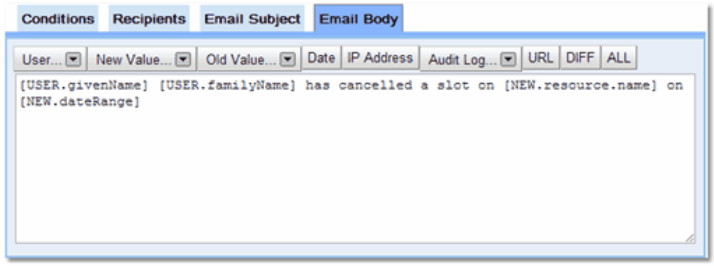
7. Add the recipients you require.

8. Enter the email's **Subject** and **Text**. Text should include the name of the *resource* and the time of the *booking*.



© 2010-2015 Exprodo Software

9. Enter the email's body



© 2010-2015 Exprodo Software

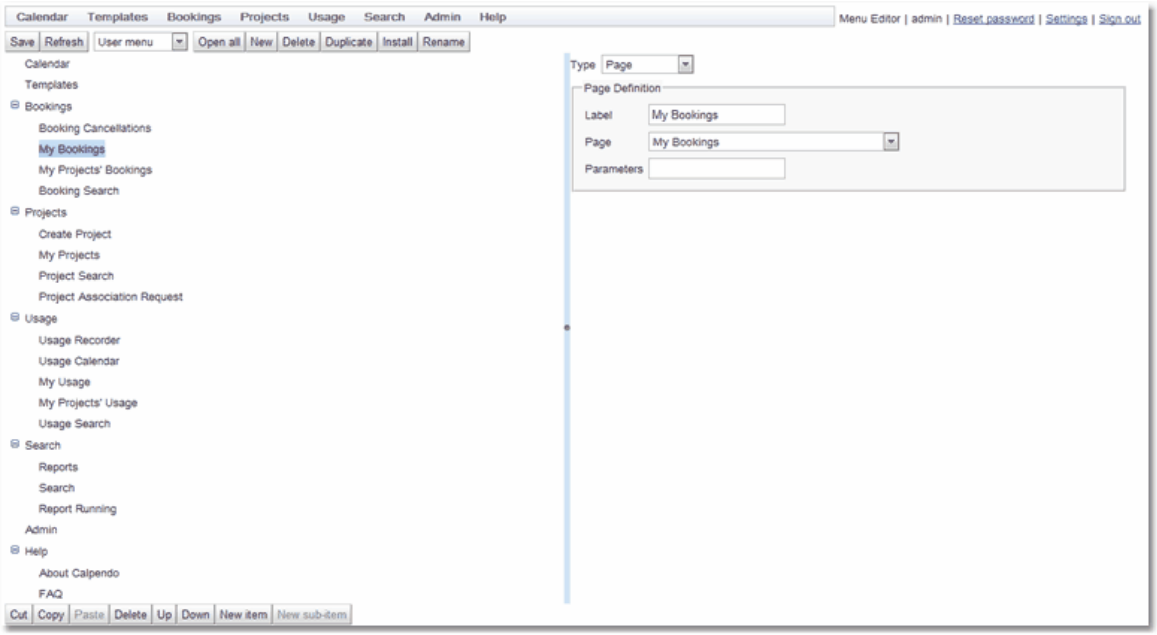
6.13 Menu Editor

Using the **Menu Editor** page the administrator can create, modify and delete the menus that exist in **Calpendo**. Once menus are created, a menu can be assigned to a user either by relying on the *global preferences* [447] for new users (see the Menus [341] tab of Global Preferences) [341] or by manually selecting the menu for particular users (see Changing A User's Settings [167]), allowing menus to be set up by the *users role* [450] or on a user by user basis. There are many different types of pages that can be added to a users menu. They include but are not limited to:

1. Customised search or *data explorer* [447] pages
2. Customised calendar [446] pages using *bookmarks* [447]
3. Pages that open up web pages external to **Calpendo**
4. Running predefined reports
5. Pages normally found only on the **Admin** menu
6. Customised *Biskit* [446] editor pages

The **Menu Editor** page consists of four parts:

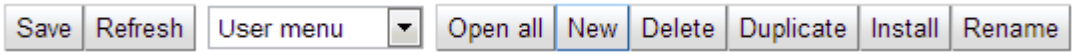
1. The menu tool bar: this is the tool bar at the top of the page, and all the buttons operate on whole menus at a time.
2. The menu item tool bar: this is the tool bar at the bottom of the page, and all the buttons operate on single items within a menu.
3. The left pane: this shows the content of the currently selected menu.
4. The right pane: this shows the detail of the currently selected menu item



© 2010-2015 Exprodo Software

The Menu Toolbar (at the top of the page)

The tool bar at the top of the page contains buttons that operate on whole menus at a time.

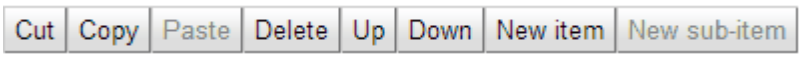


For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition 96 chapter.

Tool Bar Item	Description
Menu drop-down	Shows a list of all the menus. Select one of them to load that menu so that it can be edited.
Open all	Opens all the menu items in the current selected menu
New	Creates a new menu
Delete	Deletes the currently selected menu. Note that this doesn't take effect until the **Save** button is pressed.
Duplicate	Creates a copy of the currently selected menu. This doesn't take effect until the **Save** button is pressed.
Install	Installs the currently selected menu into the browser. This is not a permanent change. When the browser is refreshed or when the current user logs out, the menu reverts to its previous state. This allows the administrator to see exactly what a menu would look like when in use.
Rename	Renames the currently selected menu.

The Menu Item Toolbar (at the bottom of the page)

The tool bar at the bottom of the page contains buttons that operate on single menu items.

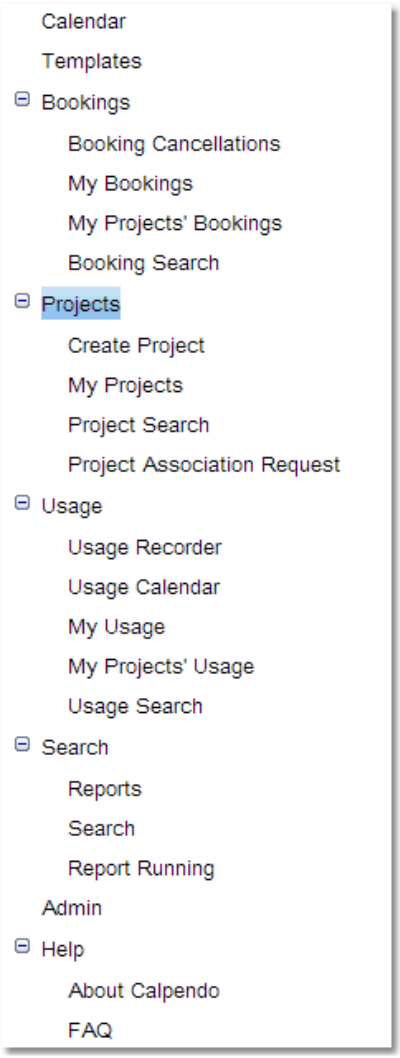


For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition 96 chapter.

Tool Bar Item	Description
Up	Moves the currently selected menu item up
Down	Moves the currently selected menu item down
New item	Creates a new menu item at the same level as the currently selected menu item.
New sub-item	Creates a new menu item as a child of the currently selected menu item.

© 2010-2015 Exprodo Software

The Left Pane



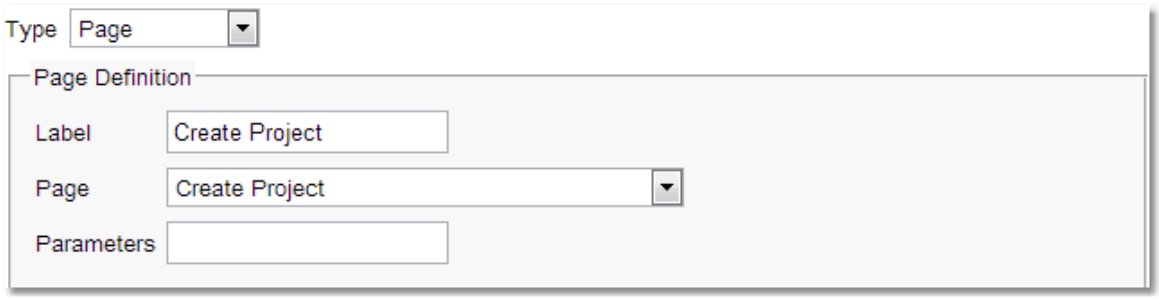
This shows a tree with the contents of the currently selected menu. If an item is selected, then:

- that menu item is loaded into the right pane so you can modify the item
- use the Menu Item Toolbar to operate on the selected menu item

© 2010-2015 Exprodo Software

The Right Pane

This shows the details of the currently selected menu item. The first *property*[448] shown is the menu item's type. There are five types, as described in this table:



Type	Description	
Page	This type of menu item is one that shows a page. When menu item is set to **Page**, the following properties need to be set:	
	Property	**Description**
	Label	This is the text that should be displayed on the menu.
	Page	This is a drop-down that allowing selection from all the different types of pages that **Calpendo** supports.
	Parameters	Some pages support additional parameters. If a page is selected that supports additional parameters, then the parameters value will be filled with a template that shows the sort of content expected.
Menu	This type of menu item is one that contains only sub-menu items. When this type is selected, also select the label that should be displayed for the sub-menu.	
Separator	Creates a horizontal line in the menu that separates the items before and after it. Once selected, also fill in the label. The label is not shown on the menu itself, but is used to display the item in the menu editor's left pane.	
Window	This represents a menu item that, when selected, will open another browser window. Set the following properties:	
	Property	**Description**
	Label	This is the text that should be displayed on the menu.
	URL	This is the address of the web page that should be loaded into the newly created browser window.

© 2010-2015 Exprodo Software

<table>
<tr><th>Type</th><th colspan="2">Description</th></tr>
<tr><td rowspan="5">Custom Page</td><td colspan="2">This type of menu item is one that shows a custom built page. After choosing Custom Page, then set the following properties:</td></tr>
<tr><td>Property</td><td>Description</td></tr>
<tr><td>Label</td><td>This is the text that should be displayed on the menu.</td></tr>
<tr><td>Token</td><td>What the configurer wishes to appear in the browsers top address bar when this menu item is selected for example in the case of the Menu Editor: #menuEditor</td></tr>
<tr><td>Custom Page Type</td><td>The type of custom page to create, there are seven possible types.
1. Biskit 446 Tree Editor
2. Bookmark Page
3. Customised Search Page
4. Data Explorer Page
5. Frame Page
6. Run Predefined Report
7. Search Results Page</td></tr>
<tr><td>Additional Information</td><td>This depends on the custom page type chosen</td></tr>
</table>

More information on the different types of **Custom Pages**.

Page	Description	Additional Information
Biskit Tree Editor	A page which which will display and edit a user-defined selection of *Biskit Types* 446. For example, if new *Biskit Types* have been created such as **Doctor** and **Nurse** and both a **Doctor** and a **Nurse** are associated with *bookings* 446, then create a menu item that takes the user to a page where they can create the **Doctors** and **Nurses**.	A list of the *Biskit Types* to be displayed on the page. Once created the order can be adjusted.
Biskit Tree Viewer Page	Provides a way to display a number of HTML pages with a navigation panel for users to choose which page to view.	1) The *Biskit Type* to be displayed 2) The *property* with the information to be viewed.
Bookmark Page	A *Calendar* page with the specified *bookmark* set up	The *bookmark* to be used.
Customised	A search page customised to your	1) The *Biskit Type* to be searched for

© 2010-2015 Exprodo Software

Page	Description	Additional Information
Search Page	specific requirements	2) The report type to be used 3) A descriptor to define additional search buttons available on the search page. See below for a definition of the descriptors allowed. 4) Any Conditions to be used in the search
Data Explorer Page	A customised *Data Explorer* page	1) The *Biskit Type* to be displayed 2) The ID of the particular record to be displayed. 0 or empty means display all 3) Whether a user viewing the custom page may change the *Biskit Type* being displayed.
Frame Page	Opens any URL as a frame inside **Calpendo**. This will display with the menu at the top of the page and then the content of the URL specified below.	The URL to be displayed. For example: url=http://www.exprodo.com
Run Predefined Report	Runs a report either system or personal.	The report to be run

The following page type may be useful when customising menus:

Page	Description	Parameters Example
Layout Editor 447	Opens up an editor which allows the user to define how to layout *properties* in a *Biskit* so that they can be read and managed easily. This is useful if your type has a large number of properties to be displayed.	
Find by URL	Opens up a page based on a URL	searchtype=

© 2010-2015 Exprodo Software

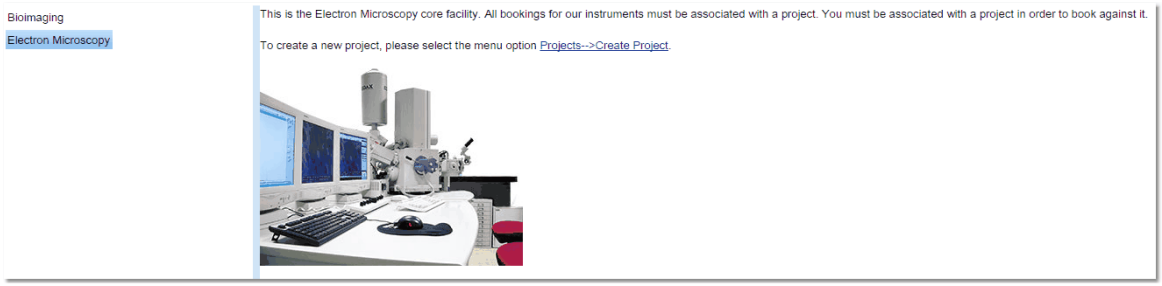
Biskit Tree Viewer

Create a custom *biskit* and give it a **Name** *property* and a **Content** *property*. The **Content** *property* should be of type **String,** and of subtype **HTML**.

Create a number of instances of the new Biskit Type, giving each a name and adding content in rich text associated with the name. This can include links and images.

Use the **Menu Editor** to add a custom page of type **Biskit Tree Viewer**, and specify the type of *biskit* and which *property* contains the **HTML** content to display.

Users then see the names of the *biskits* appear in a navigation pane on the left with the associated content on the right.


Bioimaging
Electron Microscopy
This is the Electron Microscopy core facility. All bookings for our instruments must be associated with a project. You must be associated with a project in order to book against it.
To create a new project, please select the menu option Projects-->Create Project.

Descriptors for Custom Search Pages

The definition of the format of a descriptor:

- line := {group}[;{group}]*
- {group} := [{initialFlags}[:[css-class][:label]]]{item}[,{item}]*
- {initialFlags} := (H|V|G|F|E|C)[S|L][A][T|O]
- {item} := {propPath}[:{itemFlags}][*][!]
- {itemFlags} := (S|L)
- {propPath} = a path, this may be the name of a property e.g. "status", or a path to the property e.g. "project.status"

Where:

- css-class is the CSS class to be applied to the top level
- the initial flags are:
  - flags to indicate the overall behaviour of the search:
    - H = horizontal button bar
    - V = vertical panel
    - F = Form-like container that displays each item with a heading in the line above the search item.
    - G = Growing form - displays like a form but can be hidden or shown by ticking a checkbox.

© 2010-2015 Exprodo Software

        - E = Expanding, a container showing a checkbox for each item, where clicking the checkbox will open up the search item
        - C = Captions around each of the items in the container
        - F = Form, labels above each widget
    - flags to indicate the default size of items. Note that these can be overridden by size flags placed on each item:
        - S = Small - widgets in this container should be small size by default
        - L = Large - widgets in this container should be full size by default
    - Sorting flags:
        - No flag means items are presented in the order specified and (when using the A flag, their natural order)
        - T = sort all items in this container by their text label
        - O = sort all items in this container by their bakery-assigned order
    - Miscellaneous flags:
        - A = automatically add items for all known properties not added elsewhere
- item flags define the size:
    - S = small - minimal widgets in a row
    - L = large - widgets taking full size the whole time

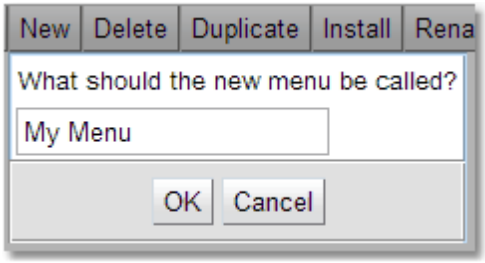
Examples:

- status:*;service.provider:*;service.location:*;{EAT}
- {CL:myCSSclass:My Label}status:*,service.provider:*;{F}service.location:*

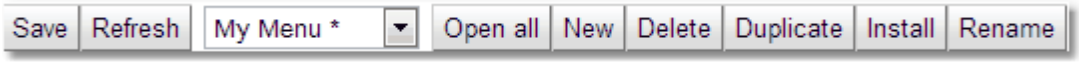
© 2010-2015 Exprodo Software

Creating A New Menu

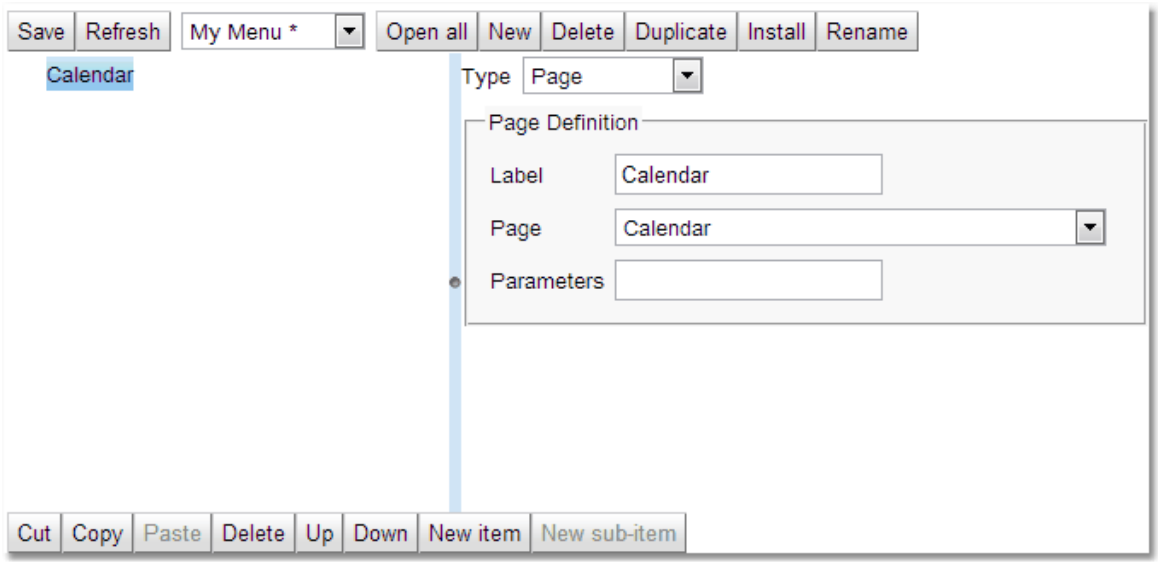
Press **New** on the menu toolbar, and type the name of the new menu into the pop up.



The name of the menu will now appear in the **Menu Toolbar Menu** drop down, it will have a * next to it to indicate it is being edited and has not been saved yet.

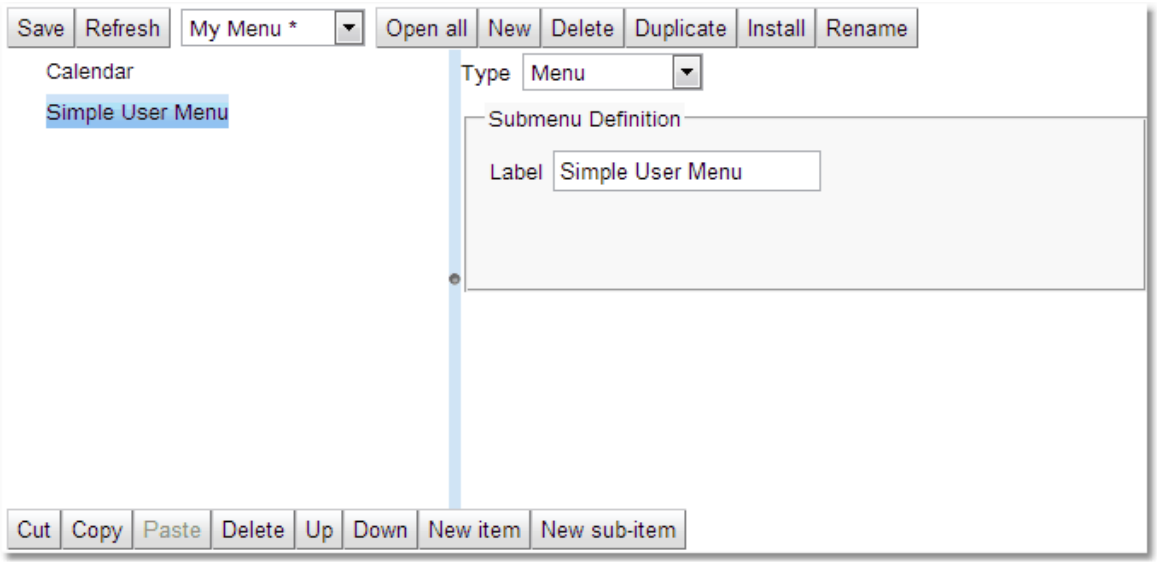


Using the **Menu Item Toolbar** at the bottom of the page, press **New Item**, then in the right pane set up the **Type** to be **Page**, and for the **Page Definition** set **Label** to **Calendar** and **Page** to **Calendar**. This will create the first menu item.

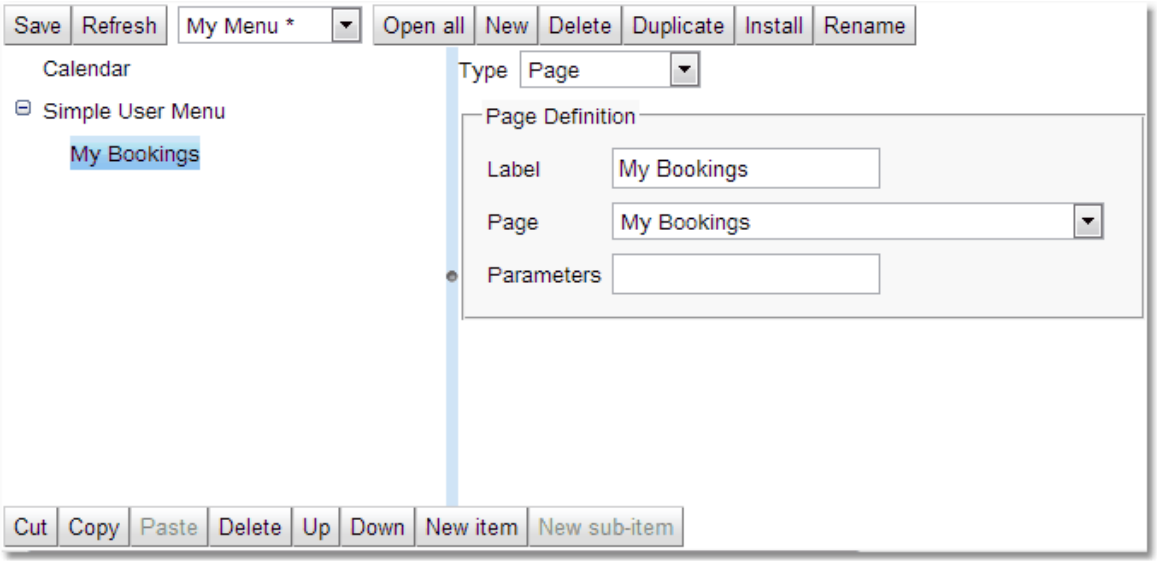


© 2010-2015 Exprodo Software

Using the **Menu Item Toolbar** at the bottom of the page, press on **New Item**, then in the right pane set up the **Type** to be **Menu**, and for the **Submenu Definition** set ***Label*** to **Simple User Menu**. This creates a simple sub menu.

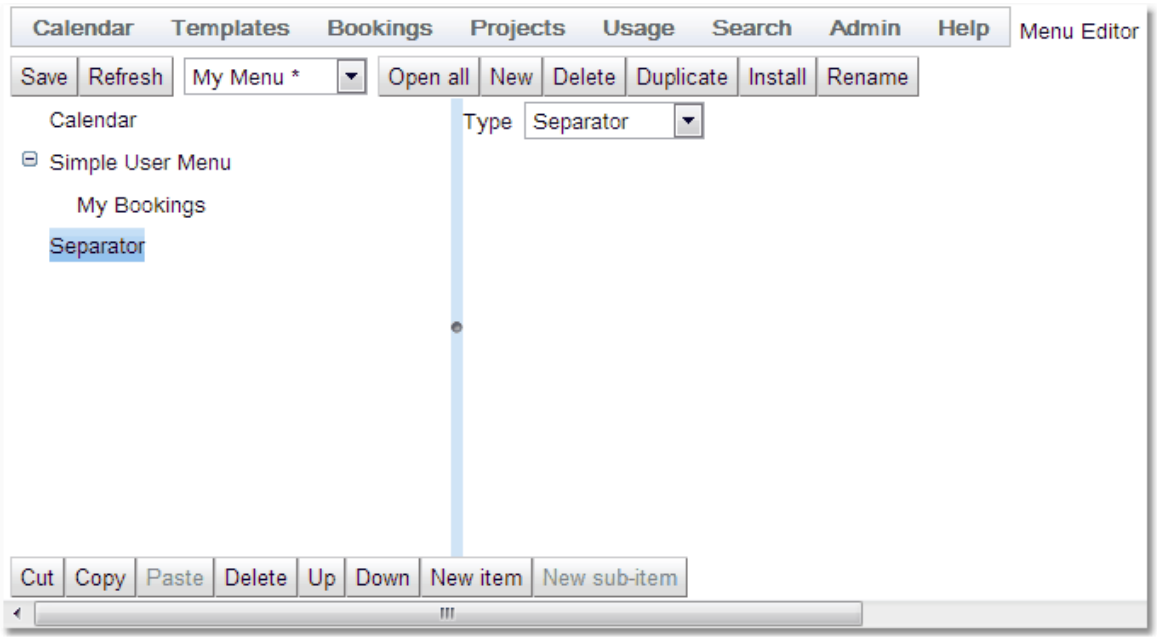


Using the **Menu Item Toolbar** at the bottom of the page, press **New Sub Item**, then in the right pane set up the **Type** to be **Page**, and for the **Page Definition** set **Label** to **My Bookings** and **Page** to **My Bookings**. This will put an item on the **Simple User Menu** sub menu.



© 2010-2015 Exprodo Software

Click on **Simple User Menu** then using the **Menu Item Toolbar** at the bottom of the page, press **New Item**, then in the right pane set up the **Type** to be **Separator**. This creates a simple separator to which is a line across the menu to split the menu into sections.



Continue adding in pages, custom pages, sub menus and separators until the menu is created. Once finished use the **Install** button on the **Menu Toolbar** to install the menu temporarily into the browser. Then check the menu to make sure it is what is required. Remember to use the **Up Down** buttons to change the order of items in the menu. When finished use the **Save** button on the **Menu Toolbar** to save the menu. The menu will still be in the menu drop down box, but will no longer have a star next to it as it has been saved. Look at the chapter on Changing a Users Settings 167 to see how to apply this menu to a particular user. Remember putting this menu as an option in Global Preferences will only change that menu for new users not existing users.

© 2010-2015 Exprodo Software

6.14 User Authentication Methods

THis section outlines how an Administrator can set up a number of different methods for user authentication, allowing **Calpendo** to be used in a single sign on environment.

6.14.1 Authentication Methods

There are three main types of authentication:

1) **Local**: The current **Calpendo** does the authentication. User names and encrypted passwords are stored in the **Calpendo** database.

2) **Internal**: These are built-in methods that allow **Calpendo** to send your user name and password to another system for authentication. **Calpendo** does not store your password in this case, but does handle it during login. The currently supported internal authentication methods include using an email server (**SMTP** or **IMAP**), any HTTP basic authentication system, or another **Calpendo**.

3) **External**: The web server that users connect to is configured to provide authentication, for example using a single-sign-on system or perhaps LDAP. In this case, **Calpendo** never sees users passwords, and authentication is solely the responsibility of the web server.

All systems automatically have **Local** authentication initialised but the administrator may also set up a number of non-local authentications and may switch off **Local** authentication. The administrator cannot switch off the authentication system they are currently logged in with, to ensure that they cannot be accidentally locked out.

Common Options For All Authentication Methods

Name	Data Type	Description
Type	String	The type of authentication method.
Name	String	The name assigned to this method. Must be unique.
Login Allowed	Boolean	Whether users can login using this method.
New User Registration Allowed	Boolean	Whether new users can register using this method. If True then new users may register for this authentication method even if logins are not allowed.

Local Authentication

Local authentication has no additional options.

IMAP Authentication and SMTP Authentication

Name	Data Type	Description
Host	String	The mail server's IP address or name.
Security	JavaEnum	Choose the type of security to be used: **None, STARTTLS, SSL/TLS**

© 2010-2015 Exprodo Software

Basic Authentication

Name	Data Type	Description
URL	String	The address of the website doing the authentication.
Realm	String	The name of the system, that is doing the authentication.

External Authentication

Name	Data Type	Description
Login Name Header	String	The name of the HTTP header that will contain the user's login name (this defaults to X-Forwarded-User)
Display Type	JavaEnum	How to display the frame that will contain the authentication handshake. If not hidden then choose its size. Choice of**: Hidden, Inline, Popup**
Display Width	String	The width of the frame within the pop up or on the login page.
Display Height	String	The height of the frame within the pop up or on the login page.

While you can use HTTP basic authentication using an internal authentication method, it's also possible to set up HTTP basic authentication using external authentication. The following shows an example of an excerpt from an Apache virtual host configuration that sets up HTTP basic authentication and also passes the REMOTE_USER setting that it generates to **Calpendo** by setting the X-Forwarded-User HTTP header. You can use any header for this, but you need to tell **Calpendo** which header to examine by setting the **Login Name Header** property. When Apache rewrites URLs, you protect /private under the rewritten URL

```
<Location "/Calpendo/com.springsolutions.calpendo.Calpendo/private">
    AuthType basic
    AuthName "Your Calpendo Realm Name"
    AuthBasicProvider file
    AuthUserFile /path/to/htpasswd/file
    Require valid-user
</Location>


RewriteEngine On

# prevent the client from setting this header
RequestHeader unset X-Forwarded-User

# see the Apache documentation on why this has to be lookahead
RewriteCond %{LA-U:REMOTE_USER} (.+)

# this actually doesn't rewrite anything. what we do here is to set RU to the match abov
# "NS" prevents flooding the error log
RewriteRule .* - [E=RU:%1,NS]
RequestHeader set X-Forwarded-User %{RU}e

# strip the REALM of Kerberos Login
# RequestHeader edit X-Forwarded-User "@REALM$" ""

RewriteRule ^/Calpendo/Service(.*) /Calpendo/com.springsolutions.calpendo.Calpendo/Service$1 [PT,L]
RewriteRule ^/Calpendo/Calpendo.html(.*) /Calpendo/com.springsolutions.calpendo.Calpendo/Calpendo.html$1 [PT,L]
RewriteRule ^/Calpendo/$ /Calpendo/com.springsolutions.calpendo.Calpendo/Calpendo.html [PT,L]
RewriteRule ^/Calpendo/(.*) /Calpendo/com.springsolutions.calpendo.Calpendo/$1 [PT,L]
ProxyPass /Calpendo http://localhost:8080/Calpendo
ProxyPassReverse /Calpendo http://localhost:8080/Calpendo
```

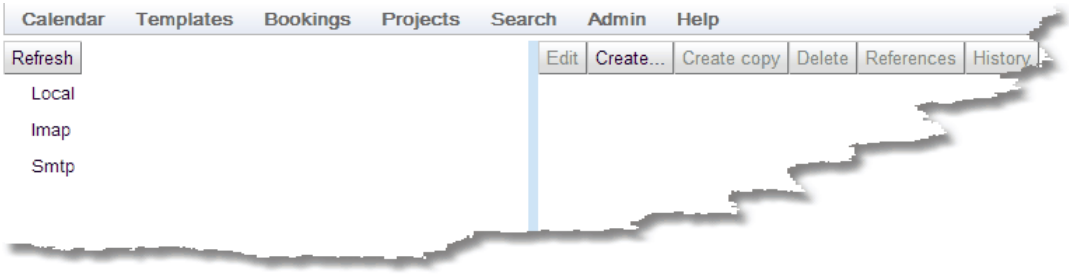
When Apache rewrites URLs, you protect /private under the rewritten URL

© 2010-2015 Exprodo Software

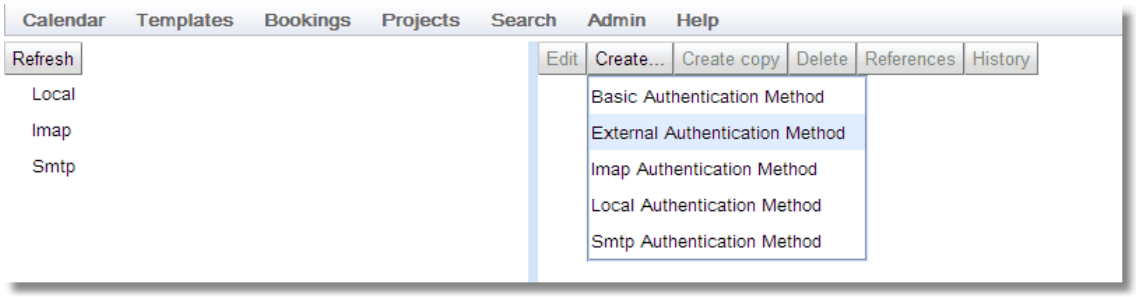
6.14.2 Authentication Methods Editor

Setting up different Authentication Methods (alternatively known as Single Sign On) is done using the **Authentication Methods Editor**, which by default is available on the menu as **Admin->Authentication Methods**.

The **Authentication Methods Editor** shows all the currently defined authentication methods and allows the administrator to create, update and delete them. This is what the looks like when first opened.

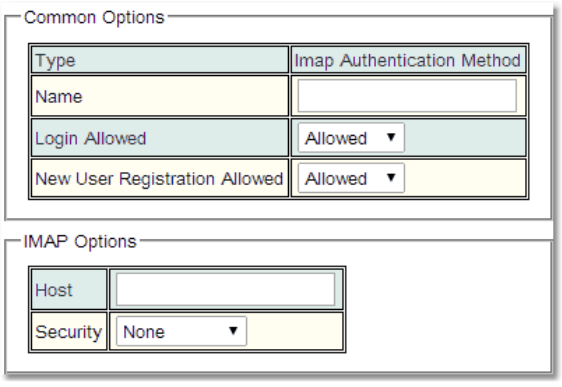


For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition 96 chapter. The only difference from the standard is that when the **Create** button is clicked a drop down menu appears to allow the administrator to choose which type of Authentication Method to be created.



After pressing **Create** or the **Edit** button to make the page editable the button bar changes:

The detail view for the authentication method will also appear, fill in the details and **Save** the method.



© 2010-2015 Exprodo Software

6.15 FAQ Editor

Calpendo provides two ways to provide help for the users specific to your facility. Firstly, it is possible to create your own documentation and add links to them on the menu. See Menu Editor[311] for guidance on how to configure the menus. The alternative is to configure the Frequently Asked Questions[133] page.

Do this with the **FAQ Editor**, which is by default on the menu as **Admin-->FAQ Editor**. This allows the creation of questions with answers, and the ability to assign them to categories. The categories provide a way to group the questions and answers. This is a way to make sure the users know who to contact in case of particular problems.

FAQ have the following properties:

Property	Description
Category	The category to place the *FAQ*[447] under
Question	The question part to the *FAQ*
Sort Order	The order this question appears in its category
Answer	The answer to the *FAQ*. See below for more information on how answers may be created.

Category has the following properties:

Property	Description
Name	The name of the category.
Sort Order	The order this category will appear in the *FAQ* page.

For more information on how the **FAQ Editor** works read the chapter on Configuring Types and Groups[266] as the **FAQ Editor** works in a similar way.

© 2010-2015 Exprodo Software

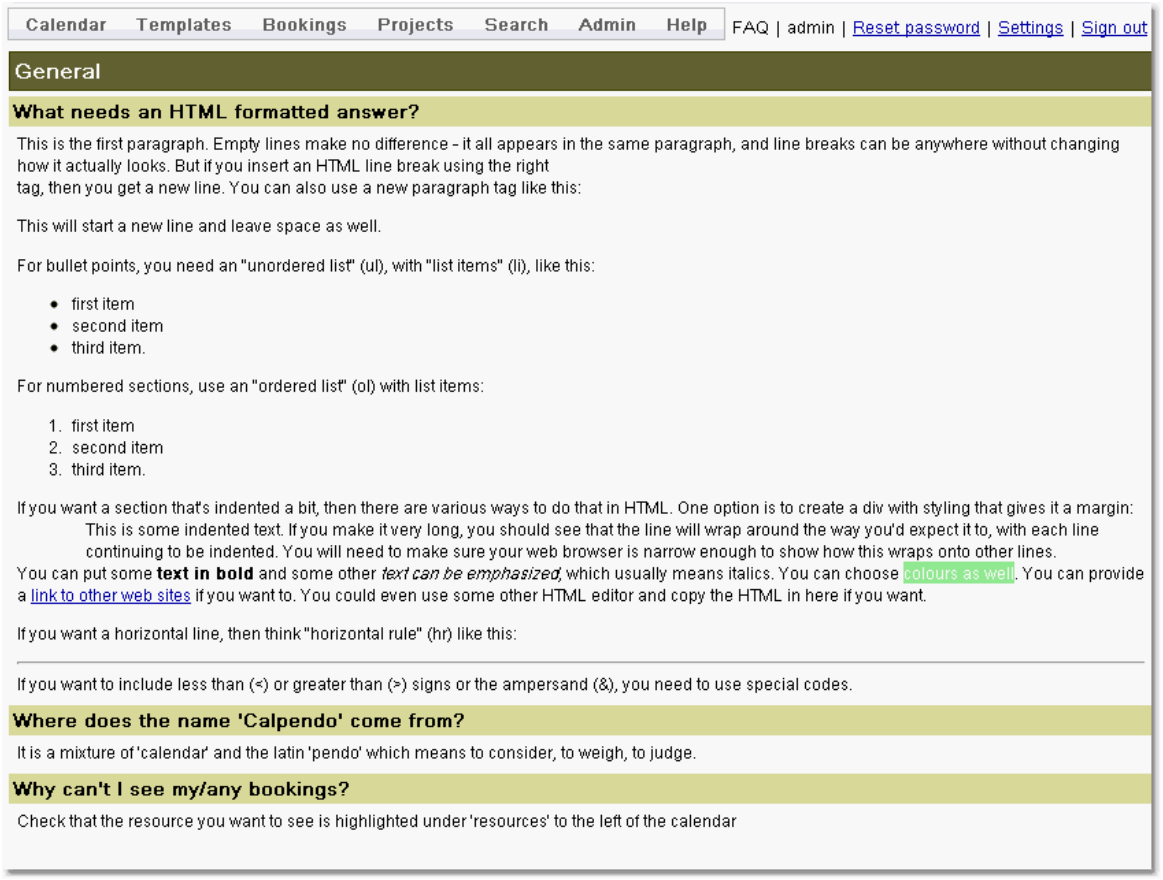
How to Configure FAQ Answers

The answers provided can contain HTML. As an example of the sort of HTML used to control the way your FAQs look, here's a sample HTML answer:

```
This is the first paragraph.

Empty lines make no difference - it all appears in the same
paragraph, and line breaks can be anywhere without changing how it
actually looks. But if you insert an HTML line break using the
right<br>tag, then you get a new line. You can also use a new
paragraph tag like this: <p>This will start a new line and leave
space as well.<p>

For bullet points, you need an "unordered list" (ul), with "list
items" (li), like this:
<ul>
  <li> first item
  <li> second item
  <li> third item.
</ul>

For numbered sections, use an "ordered list" (ol) with list items:
<ol>
  <li> first item
  <li> second item
  <li> third item.
</ol>

If you want a section that's indented a bit, then there are various
ways to do that in HTML. One option is to create a div with styling
that gives it a margin:

<div style='margin-left:50px;'>
This is some indented text. If you make it very long, you should see
that the line will wrap around the way you'd expect it to, with each
line continuing to be indented. You will need to make sure your web
browser is narrow enough to show how this wraps onto other lines.
</div>

You can put some <b>text in bold</b> and some other <em>text can be
emphasized</em>, which usually means italics. You can choose
<span style='color:white; background-color: lightGreen;'>
  colours as well
</span>.

You can provide a <a href='http://www.calpendo.com'>link to other
web sites</a> if you want to. You could even use some other HTML
editor and copy the HTML in here if you want.
<p>
If you want a horizontal line, then think "horizontal rule" (hr)
like this:
<hr>
If you want to include less than (&amp;lt;) or greater than (&amp;gt;)
signs or the ampersand (&amp;amp;), you need to use special codes.
```

© 2010-2015 Exprodo Software

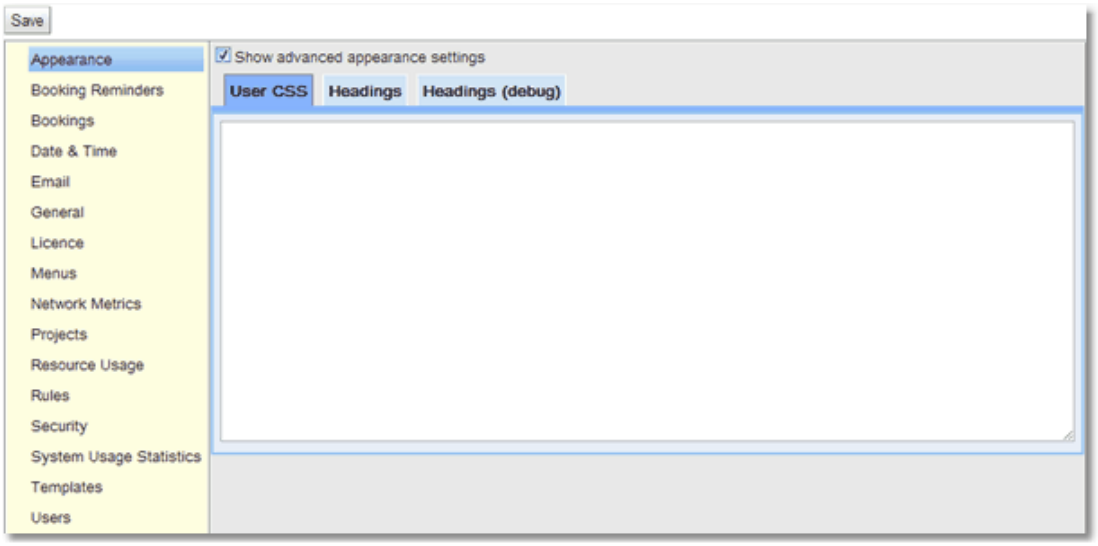
and this is the way that it looks:


Calendar Templates Bookings Projects Search Admin Help
FAQ | admin | Reset password | Settings | Sign out
General
What needs an HTML formatted answer?
This is the first paragraph. Empty lines make no difference - it all appears in the same paragraph, and line breaks can be anywhere without changing how it actually looks. But if you insert an HTML line break using the right
tag, then you get a new line. You can also use a new paragraph tag like this:
This will start a new line and leave space as well.
For bullet points, you need an "unordered list" (ul), with "list items" (li), like this:
first item
second item
third item.
For numbered sections, use an "ordered list" (ol) with list items:
1. first item
2. second item
3. third item.
If you want a section that's indented a bit, then there are various ways to do that in HTML. One option is to create a div with styling that gives it a margin:
This is some indented text. If you make it very long, you should see that the line will wrap around the way you'd expect it to, with each line continuing to be indented. You will need to make sure your web browser is narrow enough to show how this wraps onto other lines.
You can put some text in bold and some other text can be emphasized, which usually means italics. You can choose colours as well. You can provide a link to other web sites if you want to. You could even use some other HTML editor and copy the HTML in here if you want.
If you want a horizontal line, then think "horizontal rule" (hr) like this:
If you want to include less than (<) or greater than (>) signs or the ampersand (&), you need to use special codes.
Where does the name 'Calpendo' come from?
It is a mixture of 'calendar' and the latin 'pendo' which means to consider, to weigh, to judge.
Why can't I see my/any bookings?
Check that the resource you want to see is highlighted under 'resources' to the left of the calendar

© 2010-2015 Exprodo Software

6.16 Global Preferences

The **Global Preferences** page appears by default on the menu as **Admin-->Global Preferences** and specifies settings that change the way **Calpendo** operates.



Once all necessary changes have been made throughout the tabs use the **Save** button to implement them.

© 2010-2015 Exprodo Software

6.16.1 Appearance

Appearance controls the look and feel of **Calpendo**. It does this by allowing the creation of CSS to overwrite the default values. The user can also set the headings for the **Host Page** and for the **Login page**, as well as a banner to be displayed at the top of each page. The **Headings (debug)** setting will only be shown when the **Show advanced appearance** box is ticked.

Setting	Description
User CSS	To specify Cascading Style Sheets (CSS) that you would like to be used by **Calpendo**. Anything specified here will override the default values, so that you can customise the look and feel.
Headings	This sets several properties that control the look of **Calpendo**, see the table below for the headers that can be changed.
Headings (debug)	To specify the same sort of content as used for Headings above, but allows alternative values when **Calpendo** is in debug mode.

An example of User CSS, would be the following

```
#registerButton
{
display: none;
}
```

This would remove the *Register New User* button from the login page.

An exampleof User CSS for Calpendo would be to change the way **Cancelled** bookings looked:

This would draw a line through the text in the header (strike through).

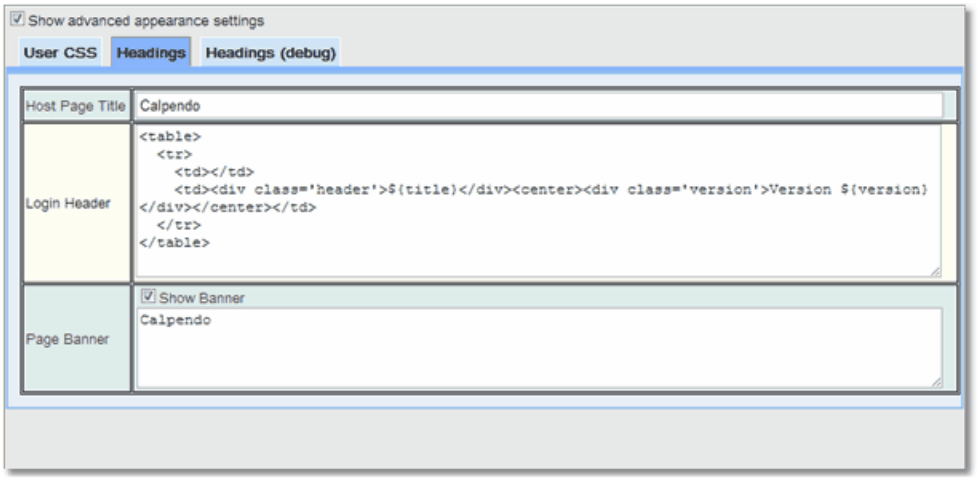
```
.Booking-CANCELLED .Booking-heading {
  text-decoration:line-through
}
```

This would draw diagonal lines through the body of the booking, these lines can have multiple colours, here they are set to be the same colour (salmon).

```
.Booking-CANCELLED .Booking-body {
 background-image: repeating-linear-gradient(-45deg, #F9966B, #F9966B 1px, transparent 1px, transparent 14px) !important;
 background-size: 20px 20px !important;
}
```

© 2010-2015 Exprodo Software

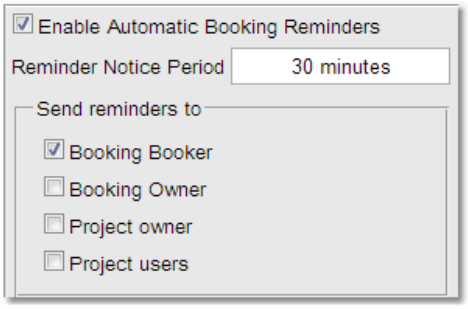
Heading options	Description
Host Page Title	This changes the HTML title of the page used for **Calpendo** so that it will show in the web browser.
Login Header	This lets the user provide alternative HTML for the header that appears on the login page.
Page Banner	If a page banner is required, this is the HTML that will be shown at the top of every page.



© 2010-2015 Exprodo Software

6.16.2 Booking Reminders

As described by Creating Bookings [52], when a *booking* [446] is made, the user can choose to enable *booking* reminders. For each user, their User Settings [32] specifies the default values for *booking* reminders. The *global preference* [447] for *booking* reminders is similar, but allows an administrator to specify the *booking* reminder user settings for newly registered users. The global setting is used only for setting up new users and nothing else.



Choose who gets a reminder email, with the options being:

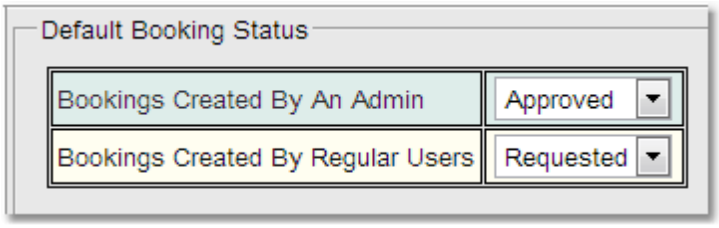
- The person that made the *booking*
- The person that owns the *booking*
- The person that owns the *booking's* *project* [448]. If there is no *project* on the *booking*, then this setting is ignored.
- The people associated with the *booking's project*. If there is no *project* on the *booking*, then this setting is ignored.

If one person falls into more than one category, for example the *project* owner is also the person that made the *booking*, they will only receive one reminder email.

6.16.3 Bookings

Default Booking Status

When a user creates a *booking* [446] with the status **Best possible**, and there are no *Time Templates* [449], *Booking Rules* [446] or *Permissions* [448], to alter that choice then the *Default Booking Status* [447] is used. This will also define the list of possibilities a user will see in the *Booking Status* [447] drop down box. If set to **Requested** only **Best Possible** and **Requested** will appear, if set to **Approved**, **Best Possible**, **Requested** and **Approved** will appear.

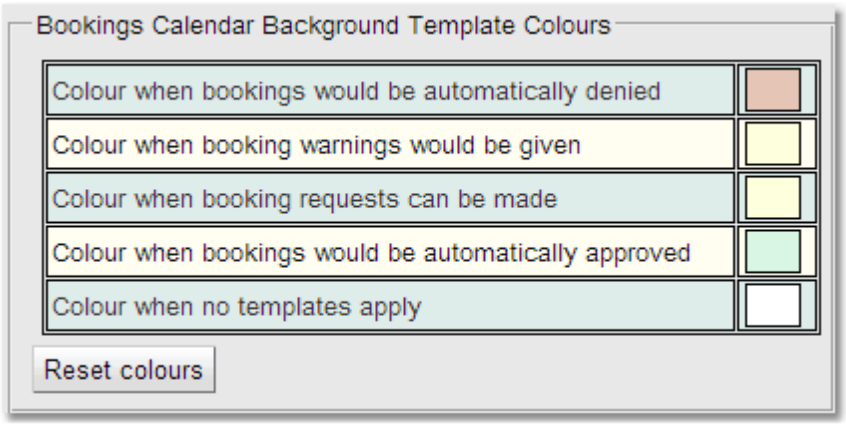


Setting	Description
Bookings Created By An Admin	Specify the default *status* for *bookings* created by a user with the **Admin** *role* [450]. This is **Approved** by default.
Bookings Create By Regular Users	Specify the default *status* for *bookings* created by users that do not have the **Admin** role. This is **Requested** by default.

© 2010-2015 Exprodo Software

Bookings Calendar Background Time Template Colours

When showing *Time Templates* on the background of the *bookings calendar*[446], different colours are used to indicate the implications of the *Time Templates* present. This is where the colours used can be setup. For each user, their User Settings[32] specifies the default colours for *Time Template* colours. The *global preference*[447] for *Time Template* colours is similar, but allows an administrator to specify the *Time Template* colour user settings for newly registered users. The global setting is used only for setting up new users and nothing else.


Bookings Calendar Background Template Colours
Colour when bookings would be automatically denied
Colour when booking warnings would be given
Colour when booking requests can be made
Colour when bookings would be automatically approved
Colour when no templates apply
Reset colours

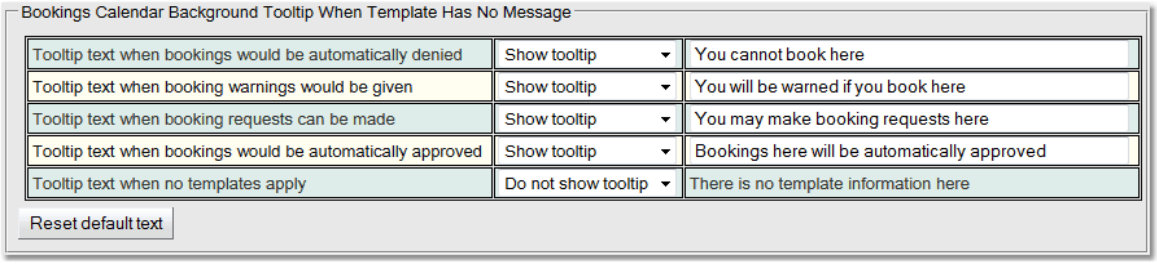
Setting	Description
Colour when *bookings* would be automatically denied	This is a light red by default
Colour when *booking* warnings would be given	This is a light yellow by default
Colour when *booking* requests can be made	This is a light yellow by default
Colour when *bookings* would be automatically approved	This is a light green by default
Colour when no *templates* apply	This is white by default

There is also a **Reset Colours** button which resets the colours back to the original defaults.

© 2010-2015 Exprodo Software

Bookings Calendar Background Tooltip When Time Template Has No Message

When displaying *Time Templates* on the background of the *Bookings Calendar*, if the user lets their mouse hover over the background then a tool tip will be shown that gives a message. If the *Time Templates* that apply at that time contain a message, then that's what will be shown in the tool tip. However, when there is no message provided, then **Calpendo** needs to know what message to show. This depends on whether the *Time Templates* indicate that a *booking* would be automatically approved, or denied etc.


Bookings Calendar Background Tooltip When Template Has No Message
Tooltip text when bookings would be automatically denied
Show tooltip
You cannot book here
Tooltip text when booking warnings would be given
Show tooltip
You will be warned if you book here
Tooltip text when booking requests can be made
Show tooltip
You may make booking requests here
Tooltip text when bookings would be automatically approved
Show tooltip
Bookings here will be automatically approved
Tooltip text when no templates apply
Do not show tooltip
There is no template information here
Reset default text

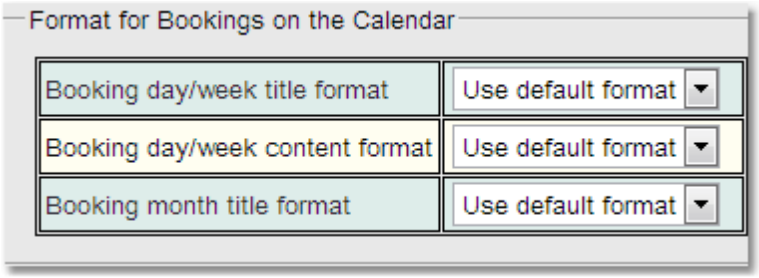
Setting	Description
Tooltip text when *bookings* would be automatically denied	Default: "You cannot book here"
Tooltip text when *booking* warnings would be given	Default: "You will be warned if you book here"
Tooltip text when *booking* requests can be made	Default: "You may make booking requests here"
Tooltip text when *bookings* would be automatically approved	Default: "Bookings here will be automatically approved"
Tooltip text when no *templates* apply	Default: "There is no template information here"

There is a button to restore the text to the original defaults.

© 2010-2015 Exprodo Software

Format For Bookings On The Calendar

When **Calpendo** displays *bookings* on the *calendar*, it extracts information from each *booking*. Specify what information it uses, and how it is laid out by providing format specifiers as described below.



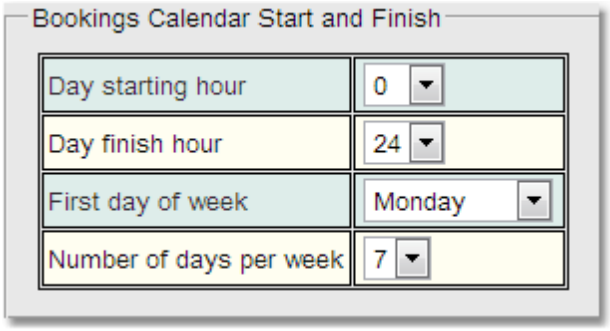
Setting	Description
Booking day/week title format	This specifies what information should be displayed in the title bar of each *booking* in the day and week view.
Booking day/week content format	This specifies what information should be displayed in the main body of each *booking* in the day and week view.
Booking month title format	This specifies the content of each *booking* in the month view.

The format for these settings is specified by providing a single line of text using a particularly complicated syntax. A future version of **Calpendo** will provide a graphical editor so that it's much easier to specify how *bookings* should be laid out on screen. For now, if you want to do this, please contact support@calpendo.com.

© 2010-2015 Exprodo Software

Bookings Calendar Start And Finish Time

By default, the *calendar* will display from midnight to midnight with a week view that contains seven days, starting on a Monday. However, these settings can be changed so that, for example, the *calender* will only show from 8am to 6pm and from Monday to Friday if your building was always closed outside those hours.



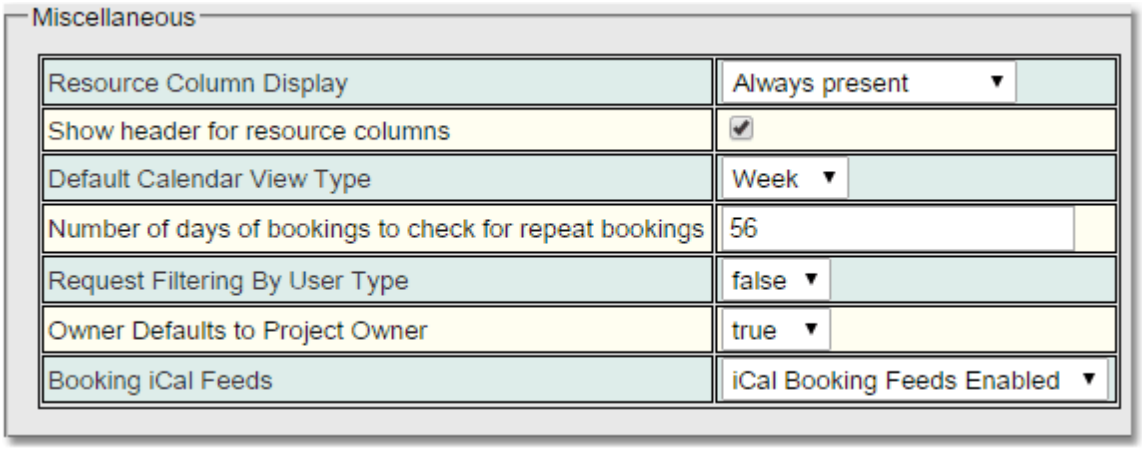
Setting	Description
Day starting hour	Default: 0
Day finish hour	Default: 24
First day of week	Default: Monday
Number of days per week	Default: 7

Miscellaneous Settings

There are various miscellaneous settings.

The first two refer to how the *resource* 449 columns will be displayed within the calendar. For each user, their User Settings 32 specifies the default values for *resource* columns. The *global preference* for *resource* columns is similar, but allows an administrator to specify the *resource* columns user settings for newly registered users. The global setting is used only for setting up new users and nothing else.

The default *calendar* view type can be overridden by a user in their User Settings 32.



© 2010-2015 Exprodo Software

<table>
<tr><th>Setting</th><th>Description</th></tr>
<tr><td>Resource Column Display</td><td>This controls the way bookings are separated into columns within each day in a day or week view. The options available are:
<table>
<tr><th>Setting</th><th>Description</th></tr>
<tr><td>Share When Possible</td><td>Bookings for different resources will be put into the same column as long as they do not occur at the same time</td></tr>
<tr><td>Separate</td><td>Bookings for different resources always appear in separate columns. If there are no bookings for a particular resource, then no column will be shown for that resource.</td></tr>
<tr><td>Always Present</td><td>Bookings for different resources always appear in separate columns, and the column will display even if there happen to be no bookings.</td></tr>
</table></td></tr>
<tr><td>Show header for resource columns</td><td>When checked, the bookings calendar will show column headings for each day in the calendar, showing the name or names of the resource(s) displayed in that column.</td></tr>
<tr><td>Default Calendar View Type</td><td>This selects whether to display a Day, Week or Month view by default when Calpendo first starts.</td></tr>
<tr><td>Number of days of bookings to check for repeat bookings</td><td>When Booking Rules and Time Templates check a booking to see if it's valid, it needs to handle repeat bookings specially. The repeat 449 instances are expanded and each instance examined separately. However, since repeats can continue forever, Calpendo cannot check them all. So this setting lets you specify how far into the future Calpendo should expand repeat bookings when checking its Booking Rules and Time Templates.

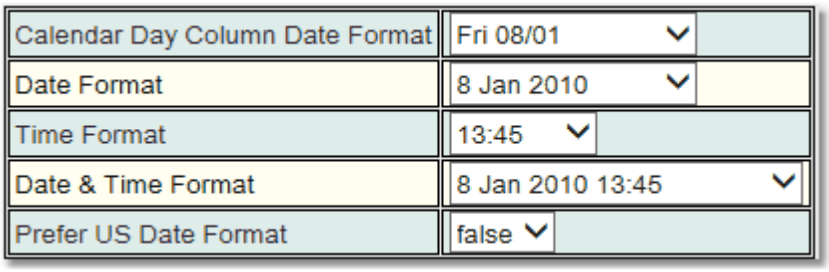
The default is 56 days.</td></tr>
<tr><td>Request Filtering By User Type</td><td>Indicates whether the booking requests page should only show requests made by or owned by users of the same type as the current user. This is false by default, so that booking requests are unfiltered.</td></tr>
<tr><td>Owner Defaults to Project Owner</td><td>When entering a booking in the calendar, the booking's owner 448 will be given a default value. Setting Owner Defaults to Project Owner to true makes the booking's owner default to the project's 448 owner (if there is a project), and setting it to false makes the booking's owner default to the user that makes the booking (the booker 446).</td></tr>
<tr><td>Booking iCal Feeds</td><td>The administrator can determine whether users can get iCal feeds for their calendars or not. If enabled an iCal button appears in the button bar above the calendar.</td></tr>
</table>

© 2010-2015 Exprodo Software

6.16.4 Date And Time

The **Date And Time** tab of the **Global Preferences** page specifies formats to use for dates and times within **Calpendo**. These are used in a number of places:

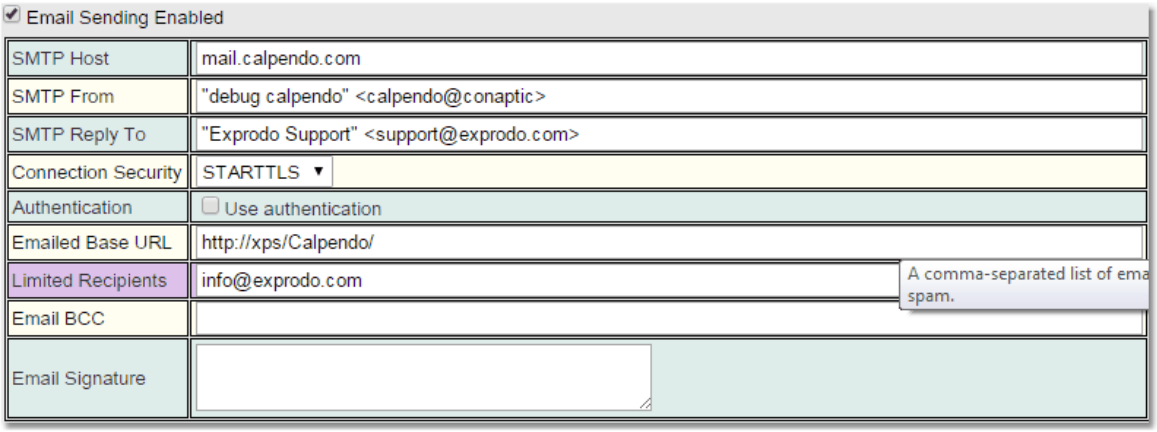
1. In Automatic Emails sent out by the server that contain and dates or times
2. When a new user is registered, the date and time formats of their user settings [32] is copied from the *global preference* [447]
3. The top of a column of the calendar views



© 2010-2015 Exprodo Software

6.16.5 Email

The **Email** tab of the **Global Preferences** page specifies the following:

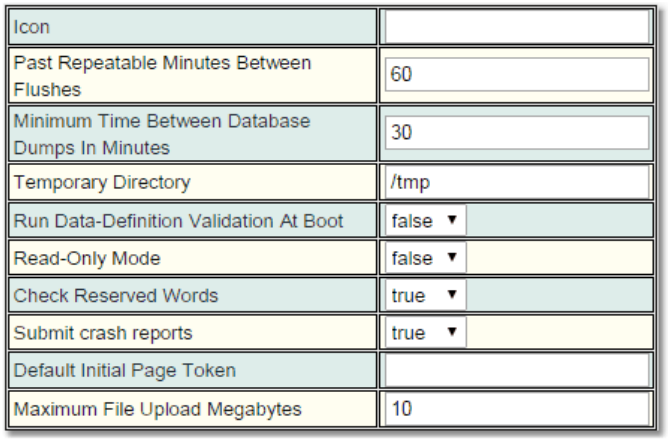


Setting	Description
Email Sending Enabled	Set this to true to allow **Calpendo** to send emails, and false to disable all outgoing emails.
SMTP Host	This is the hostname of the SMTP server to be used for sending outgoing email.
SMTP From	This is the name, and possibly email address, that outgoing emails should appear to come from. For example, for a **Calpendo** running at example.com, choose a value like: "Example Calpendo" <Calpendo@example.com>
SMTP Reply To	The reply to address on any email that is sent out. Set so that if users do reply to an email it goes to someone rather than getting an error message.
Connection Security	Choose the security level that should be used for the connection between your SMTP email server and the **Calpendo** server.
Authentication	Select whether to use authentication for the connection between your SMTP email server and the **Calpendo** server, and if so, select the user name and password that should be used by **Calpendo** to connect.
Emailed Base URL	This is the URL that should be used to get to **Calpendo** and is used by Automatic Emails 297 that reference a URL 304. It is useful when sending an email that includes a direct link to a page within **Calpendo**, and the main part of the URL needs to be injected automatically.
Limited Recipients	During testing anything related to email, restrict the email addresses that are allowed to receive email from **Calpendo**. Provide a comma-separated list of email addresses to restrict outgoing email to go only to those addresses. If this field is left empty, then email will be allowed to go to anybody.
Email BCC	This is a comma-separated list of email addresses that should receive a copy of all outgoing emails from **Calpendo**. This is useful for checking that **Calpendo** is sending the expected emails, particular while **Calpendo** is being configured.
Email Signature	This is a signature that is included at the end of every outgoing email. If left empty, then a default signature will be added.

© 2010-2015 Exprodo Software

6.16.6 General

The **General** tab of the **Global Preferences** page provides the following:



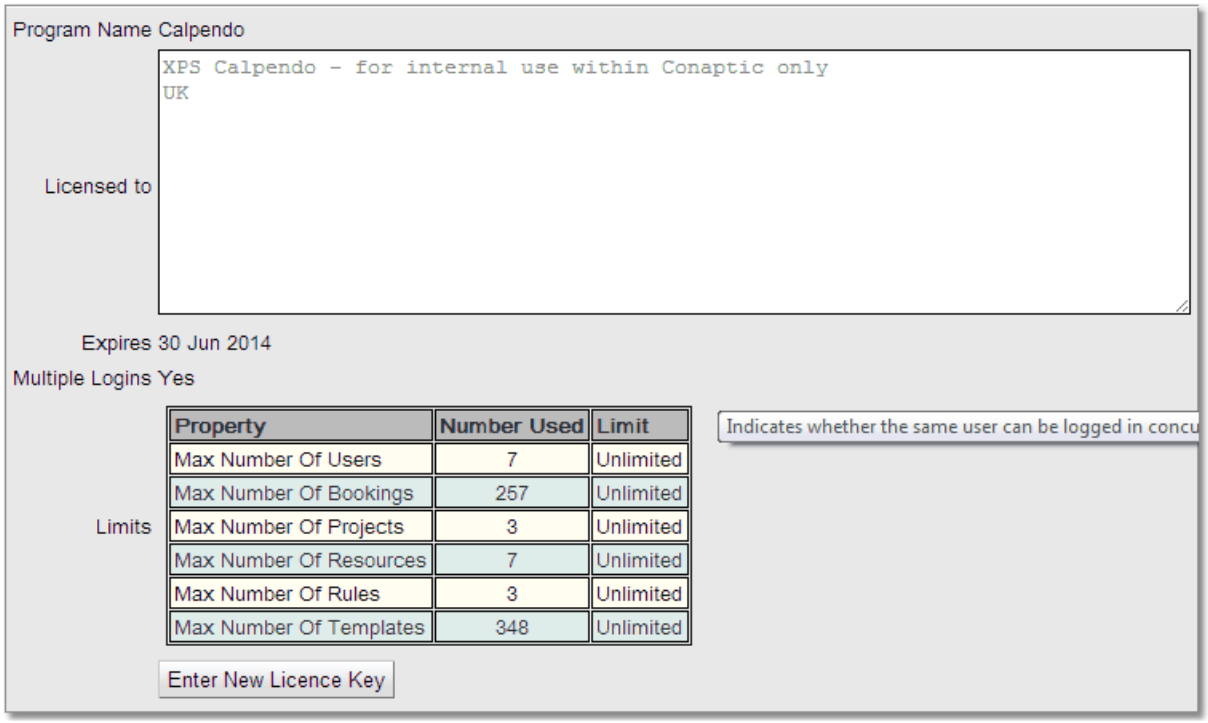
Setting	Description
Icon	This provides the name of an image file that should be used to display on the *Bookings Calendar*[446] and (using the default settings for the appearance headings[329]) on the login page.
Past Repeatable Minutes Between Flushes	**Calpendo** has some *Biskit Types*[446] that support repeats such as *Events*[447], *Bookings*[446] and *Time Templates*[450]. As the time for each instance of the *repeat*[449] passes, a new *Booking* or *Event* etc. is created to represent the instance of the repeat that happened. This is done so that, when a repeat *Booking* (or *Event* etc) is changed, you do not change the record of what happened in the past. This setting sets up how often **Calpendo** looks for *repeats* that have just happened and need a new historical entry creating.
Minimum Time Between Database Dumps In Minutes	This limits the frequency with which database dumps can be created. This is here because database dumps take some time and can degrade system performance, so they don't want to be allowed to be run too often.
Temporary Directory	This is the name of a directory that is used to store temporary files that are created for downloading.
Run Data-Definition Validation At Boot	If *Biskit definitions*[446] are changed using the Bakery[351], then it is possible that the system may be left in an inconsistent state. This setting disables the boot-time checks that are done to make sure the *Biskit definitions* are consistent. If there is a problem and boot-time checks are enabled, then **Calpendo** will not boot properly. If there is a problem and boot-time checks are disabled, then the user may be able to correct the problem in the Bakery[351], but the issue may cause something to work incorrectly in the meantime.
Read-Only Mode	This indicates whether **Calpendo** should operate in a read-only mode.
Check Reserved Words	Choose whether to enable protection that stops users from using reserved database words for table, column and property names when using the Bakery[351]. This is particularly important if exporting the **Calpendo** database to a different type of database.
Submit Crash	Choose whether to allow crash reports to be emailed to **Exprodo**

© 2010-2015 Exprodo Software

Setting	Description
Reports	**Software**. Automatic delivery of crash reports are a very useful tool for helping to find and fix problems and are only used for this purpose.
Default Initial Page Token	Choose which page is displayed by default when this program is first started. Enter the text after the '#' in the browser's address bar, and before any non-alphanumeric characters
Maximum File Upload Megabytes	choose the maximum size of a file to be uploaded as an attachment. The default is 10 Megabytes.

6.16.7 Licence

This tab lets the administrator enter a licence key as well as see what the current licence key limits are. When provided with a new key, click the **Enter New Licence Key** button and copy and paste the new licence into the **Enter New Licence Key** text box, and press **OK**. Once you have been returned back to this view make sure you save the changes using the **Global Preferences Save** button.

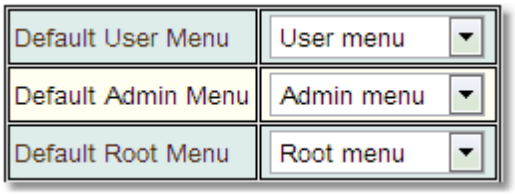


The table at the bottom of the **Licence** area will specify how many of each of the *properties* 448 are being used and the limit defined by the current key.

© 2010-2015 Exprodo Software

6.16.8 Menus

When a new user logs in to **Calpendo** for the first time, they are assigned a menu. This can overridden by specifying a menu, as described in Changing A User's Settings[167]. However, if *settings*[449] for a new user have not been set up, then the menu they are assigned depends on their *user roles*[450], and this is controlled by the **Menus** tab of the **Global Preferences** page:



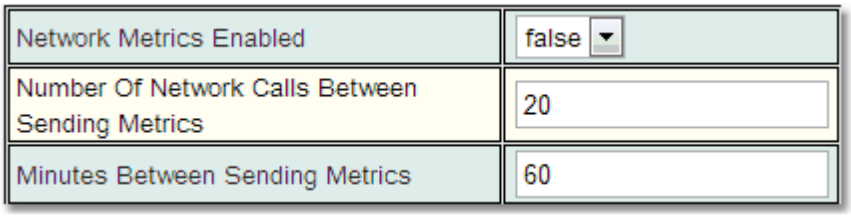
Setting	Description
Default User Menu	This is the menu to be assigned to users that have neither the **Admin** nor the **Root** *role*.
Default Admin Menu	This is the menu to be assigned to users that have the **Admin** *role* but not the **Root** *role*.
Default Root Menu	This is the menu to be assigned to users that have the **Root** *role*.

© 2010-2015 Exprodo Software

6.16.9 Network Metrics

Calpendo can keep information that tracks how long each network call took. This information can be turned on if you want to investigate network performance, but it should normally be left turned off.

The **Network Metrics** tab of the **Global Preferences** page provides the following:



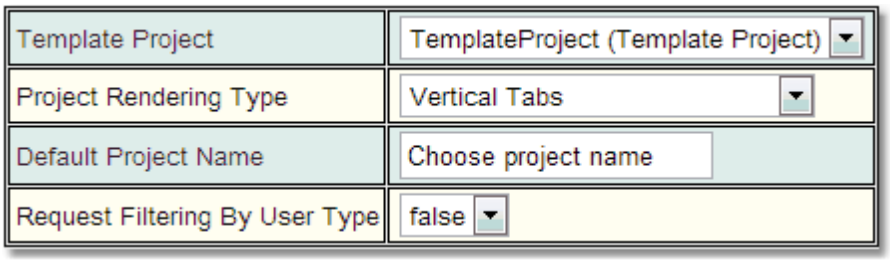
Setting	Description
Network Metrics Enabled	If this is set to *true*, then *network metrics*[448] will be kept, otherwise they will not. Note that by setting this to false, statistics that are already in the database will not be affected. Only new information will be affected by this setting. **Enabling Network Metrics** will slow down Calpendo a little.
Number Of Network Calls Between Sending Metrics	Indicates how many network calls each web browser much make before sending its *network metric* information to the server. The larger this number is, the fewer network calls will be created as an overhead. However, larger numbers increase the greater the memory requirements and also increase the amount of data that will be lost when the user closes their browser. This is because outstanding information collected by the browser but not sent to the server is not sent when the browser it closed.
Minutes Between Sending Metrics	This is the period of time between sending *metrics* to the server. This is only used during periods of low activity when you have some information to send to the server, but you are not active enough to reach the **Number Of Network Calls Between Sending Metrics**.

If *Network Metrics* are enabled, the data will be stored in the database. In order to access the data use Search[108]. Once in the **Search** page set the **Search** *Biskit Type*[446] to **Network Metric**, as there could be a lot of information to be returned make sure Conditions...[98] are used to set at least time limits for the information returned. For more information read the chapter on using Search[108].

© 2010-2015 Exprodo Software

6.16.10 Projects

The **Projects** tab of the **Global Preferences** page allows the following to be set up:



Setting	Description
Template Project	This specifies the *project*[448] that should be used as a *template*[449] for newly constructed *projects*. See Project Template[206] for more details.
Project Rendering Type	When configuring project properties[206], a group name can be assigned to each *property*[448]. Then, when a *project* is displayed, its *properties* are shown in these groups according to the group name set on each *property*. Normally, a *project* displays its *properties* with all the *property* groups visible all at once. However, if a large number of *properties* and groups are to be created, then it may be better to display those *properties* in separate tabs, with one tab per group. Then choose the **Project Rendering Type** to select the way that the different *property* groups are displayed. The options to use are:

Setting	Description
Vertical Tabs	Tabbed vertically, useful for a large number of tabs.
Vertical Tabs (sorted by name)	As above but sorted by tab name.
Horizontal Tabs	Tabbed horizontally.
Captions	All properties visible at the same time, arranged by groups. Below is an example of project information displayed in caption format.

Default Project Name	When somebody creates a new *project*, the new *project* will initially be copied from the **Template Project**, but then its status will be changed to **Requested**, and its name will be changed to whatever is specified here for **Default Project Name**.
Request Filtering By User Type	If this is set to true, then the *project* requests page will only show *project* requests from users with the same *user type*[450] as the currently logged in user. Otherwise, all *project* requests will be shown.

© 2010-2015 Exprodo Software

6.16.11 Resource Usage

The **Resource Usage** tab of the **Global Preferences** page allows set up of the items below:

Miscellaneous

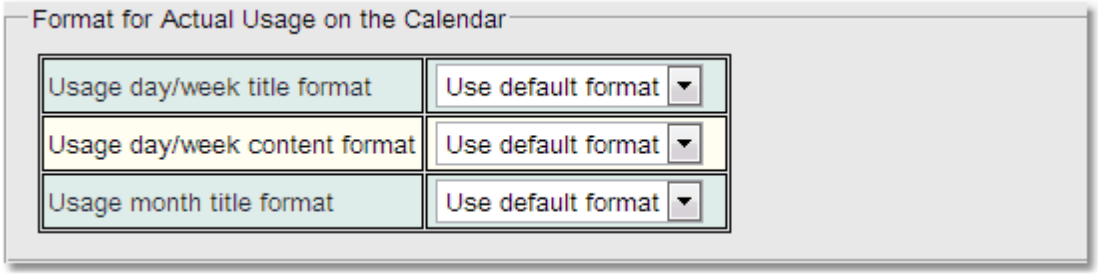
These are some miscellaneous set up for the *Resource Usage Calendar*[449].

Setting	Description
Usage Outcome When All Went Well	The potential *resource usage*[449] outcomes can be customised, but one of them will be used when the user says that everything went well with their session. This setting chooses which outcome is the one recorded when they say it went well.
Resource usage calendar future background colour	Users cannot enter resource *usage* in the future because it doesn't make sense. This setting allows chooses the colour that will be displayed in the background of the *Resource Usage Calendar* for future dates.

© 2010-2015 Exprodo Software

Format For Actual Usage On The Calendar

This allows selection of the format for items displayed on the *Resource Usage Calendar*.


Format for Actual Usage on the Calendar
Usage day/week title format
Use default format
Usage day/week content format
Use default format
Usage month title format
Use default format

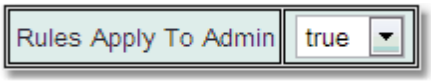
Setting	Description
Usage day/week title format	This specifies what information should be displayed in the title bar of each resource *usage* in the day and week view.
Usage day/week content format	This specifies what information should be displayed in the main body of each resource *usage* in the day and week view.
Usage month title format	This specifies the content of each resource *usage* in the month view.

Note

The format is specified by providing a single line of text using a particularly complicated syntax. A future version of **Calpendo** will provide a graphical editor so that it's much easier to specify how *usage* should be laid out on screen. For now, if this is required, please contact support@calpendo.com.

6.16.12 Rules

The **Rules** tab of the **Global Preferences** page sets up the following:



Setting	Description
Rules Apply To Admin	This chooses whether or not Booking Rules apply to users with the **Admin** *role*[450].

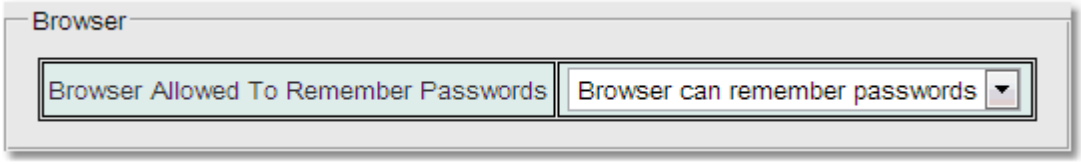
© 2010-2015 Exprodo Software

6.16.13 Security

The **Security** tab of the **Global Preferences** page sets up the items below:

Browser

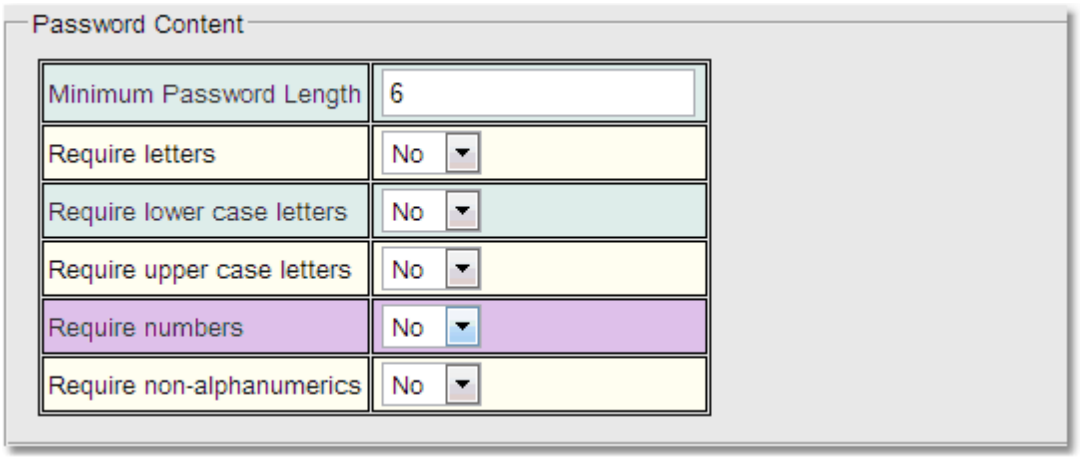
A user can specify that their browser will remember logins and passwords. On public systems this may not be a good idea, so you can disable this facility.



Setting	Description
Browser Allowed To Remember Passwords	**Calpendo** is configured so that it allows the user's browser to remember their login name and password (if this facility is enabled by each user's browser). Disable this facility at a global level by setting this to **false**.

Password Content

This provides control of the content of passwords that people are allowed to use.

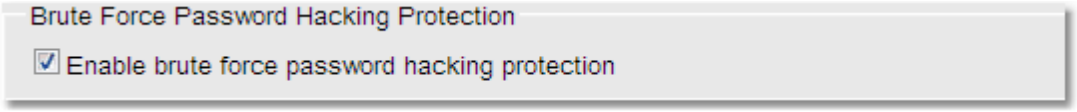


Setting	Description
Minimum Password Length	This specifies how many characters must be in each password. A user will not be able to set a password that is shorter than this setting.
Require lower case characters	Specifies whether passwords must include lower case characters.
Require upper case characters	Specifies whether passwords must include upper case characters.
Require numbers	Specifies whether passwords must include numbers.
Require non-alphanumerics	Specifies whether passwords must include characters that are not numbers or letters.

© 2010-2015 Exprodo Software

Brute Force Password Hacking Protection

This enables or disables protection against brute force password hacking attempts. This should probably always be enabled. The only potential problem with this is that the protection involves disabling logging in from the IP address being used by the hacker. So if there is a hacker on your own network, it's possible that this protection could stop some legitimate users logging in temporarily. This is only ever temporary.

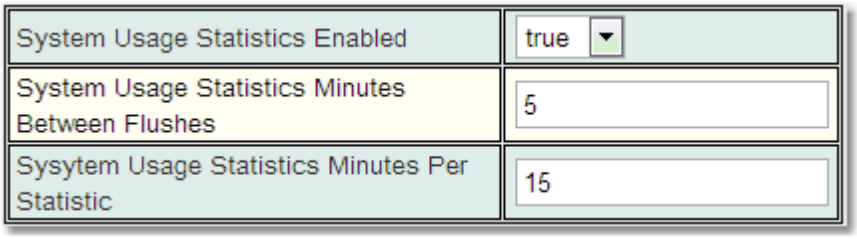


© 2010-2015 Exprodo Software

6.16.14 System Usage Statistics

Calpendo can keep *statistics*[449] that track how much each user has been using it. *Statistics* are kept by collating user activity into periods of time, so that **Calpendo** can record how much activity there was in each block of time. For each user that has been active in a given block of time, a *statistic* is stored in the database.

The **System Usage Statistics** tab of the **Global Preferences** page lets you provide the following:



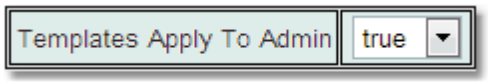
Setting	Description
Usage Statistics Enabled	If this is set to **true**, then *usage statistics* will be kept, otherwise they will not. Note that by setting this to **false**, *statistics* that are already in the database will not be affected. Only new *statistics* will be affected by this setting. **Enabling Usage Statistics** will slow down **Calpendo** a little.
Usage Statistics Minutes Between Flushes	This specifies how often **Calpendo** should flush *statistics* out to the database. For performance reasons, this should not be made too small.
Usage Statistics Minutes Per Statistic	This indicates how long each block of time is, during which *statistics* are put together. The larger this value is, then the less space will be taken in the database, and the lower the time resolution of the *statistics*.

If **System Usage Statistics** is enabled, the data will be stored in the database. In order to access the data use Search[108]. Once in the **Search** page set the search *Biskit Type*[446] to **System Usage**. As there could be a lot of information to be returned make sure you use Conditions...[98] to set at least time limits for the information returned. For more information read the chapter on using Search[108].

© 2010-2015 Exprodo Software

6.16.15 Time Templates

The **Templates** tab of the **Global Preferences** page sets up the following:

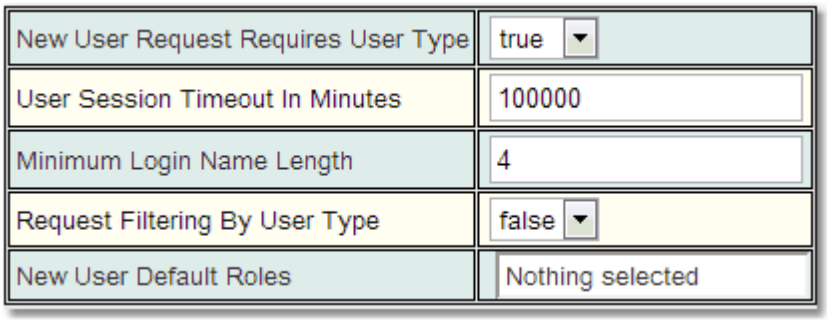


Setting	Description
Templates Apply To Admin	This chooses whether or not *Time Templates*[449] apply to users with the **Admin** *role*[450].

© 2010-2015 Exprodo Software

6.16.16 Users

The **Users** tab of the **Global Preferences** page sets up the following:



Setting	Description
New User Request Requires User Type	If this is set to true, then whenever a new user registers with **Calpendo**, they must specify their *user type*[450]. See Configuring Types And Groups[266] for more information about types.
User Session Timeout In Minutes	This indicates how long a user's session will remain valid without any activity. Once this time has passed without them using **Calpendo**, then they will have to log in again. This should not be zero or negative. If a user is timed out, any request they make at the point of the time out will be automatically resubmitted when they log back in.
Minimum Login Name Length	This specifies how many characters must be in each user name. A user will not be able to register with a login name that is shorter than this setting.
Request Filtering By User Type	If this is set to true, then the user requests page will only show user requests from users with the same *user type* as the currently logged in user. Otherwise, all user requests will be shown.
New User Default Roles	When a new user is created specify the *roles*[450] they can have by default. See User Roles[150] for more information.

© 2010-2015 Exprodo Software

6.17 Bakery

Calpendo is built on Exprodo, a web-based system that is designed to be able to display and edit the contents of a database. To be able to do this, **Exprodo** allows the building of definitions of the data in the database. In other words, while **Calpendo** has a built-in set of data types that it handles,the user can add extra data types and modify the predefined data types by adding new properties and change the way existing properties are displayed.

As well as being able to create definitions of the data in a database so that the user can see and edit the database contents, **Exprodo** provides additional facilities to help with interaction with that data:

- Automatic Emails 297 to be sent in response to data being changed
- Customisable Menus 311
- A Permissions 278 system based on very adaptable Conditions 98
- Dynamic reporting tools.

Each of these facilities makes heavy use of the fact that the data types themselves are defined in the database. It lends itself to providing a very flexible system. Therefore, the very heart of the system is the means by which one can view and modify the definitions of the data types, and that is precisely what the *Bakery* 446 does.

The data types themselves are called *Biskit Types* 446, and the definition of the content of a *Biskit Type* is known as a *BiskitDef* 446. Each *BiskitDef* defines certain information, such as its name, the labels used to display that name, the database table that contains the *Biskit Type*, and how to display the name of its *Biskits* 446 *(*individual occurrences of the *Biskit Type.* i.e. User is a *Bisket Type*, Joe Bloggs would have a *Biskit* in the database defining his information as a specific occurrence of the User *Bisket Type*, with the *properties* 448 of that *Biskit* filled in with his specific information.). It also stores a list of *property* definitions, with each *property* definition defining its *Biskit Type*, how it's stored in the database, how to display the label for the property, the tool tip text to be displayed when the mouse hovers over each *property* value and a number of other aspects of the *property*.

When saving *Biskit Types* and *properties*, **Calpendo** does check that reserved words have not been used for column names, table names, *Biskit Types* and *property* names. There are currently over eight hundred words such as Select, Into, From, Start, Full which are reserved. The system administrator can switch this checking off in Global Preferences, General 339 section, but this is not advisable.

The *Bakery* allows these things to be customised, as well as adding the users own *BiskitDefs* so that **Calpendo** can be used to interact with the users own data.

6.17.1 Property Storage Mechanisms

A database has many tables, with each table having many columns. When a new *Biskit* 446 is created, a new row is added to the *Biskit's* table, containing its *properties* 448. When a new *property* is added to a *Biskit* in **Calpendo**, one would normally also add a new column to the appropriate database table. However, modifying tables like this is not something that all users would be comfortable doing.

© 2010-2015 Exprodo Software

Calpendo helps with this by creating table columns automatically, but it will not remove unused columns, and it will not change the *Biskit Type*[446] contained in a column. This provides some protection against accidental destruction of data while changing the *Biskit* definitions, but also means that getting the configuration right does have some difficulties. This is particularly true if there needs to be a change of the type of data stored by a *property*, which has to be handled by creating a new column for it.

However, there is an alternative that does not involve creating a new table column when adding a new *property*. **Calpendo** supports what are called **Indirect** properties which are stored in a separate table in such a way that new *properties* do not require changing the table structure.

In all, there are three *property* storage mechanisms: **Static**, **Dynamic** and **Indirect**. This table compares them:

Comparison	Static Properties	Dynamic Properties	Indirect Properties
Description	Built-in to **Calpendo**	Created using the *Bakery*[446]	Created using the *Bakery*
Structure	Each *property* in a column in the *Biskit's* main table	Each *property* in a column in the *Biskit's* main table	One row added to the *Biskit's* indirect *properties* table for each indirect *property*.
Adding new property	Not possible without a new version of **Calpendo**	• Add in the *Bakery* • Ask the *Bakery* to update the database schema • Tell **Calpendo** server to reload the data definitions when you have finished making all changes	Add in the *Bakery*
Advantages	• Efficient storage and retrieval • The fastest access mechanism	• Efficient storage and retrieval • Not quite as fast as static properties	• Does not require table structure to be changed to add a new indirect *property* • Harder to make mistakes and get confused
Disadvantages	Cannot be changed without new version of **Calpendo**	Need to be aware of database columns and their *Biskit Types*	Much less efficient storage and retrieval than the other property types

Users of **Calpendo** cannot create or delete **Static** *Biskits* or *properties* but can change some of the *meta properties*[448] of both the *Biskit* and the *properties*, such as the labels, tool tips, whether required and default values.

Users should be using dynamic *properties* rather than indirect *properties*, unless they really do not wish to change the **DB Schema**, and are happy with the speed of the indirect *properties*.

© 2010-2015 Exprodo Software

6.17.2 Biskit Definitions

It can be a bit confusing to talk about definitions of *properties*[448]. So let's start by using an example, and looking at the definition of the **Session** *Biskit*[446] that is used to store information about who is using **Calpendo**. A **Session** has the following *properties*:

Name	Data Type	Description
user	*Biskit*	The user whose session this is
started	Date Time	The date/time when this session started
lastUsed	Date Time	The date/time when this session was last used
IPAddress	String	The IP address of the user's computer
sessionID	String	An ID used to represent the session when the user's web browser sends requests to the server
userAgent	String	The web browser's reported user agent information. This tells you which web browser they are using.

Now, the *BiskitDef*[446] for a **Session** must contain information about each of the above six *properties*. For each one, it has to know the Data Type, its name, the label used to display it and so on. However, the **Session** *BiskitDef* must also store *meta-properties*[448] of **Session** itself, such as its type ("Session"), its database table name and the text used to display the name "Session" in a number of contexts.

There are quite a few different types of property that a *Biskit* can store, and these are described in the next section, Property Definitions[358]. The *meta-properties* of a *Biskit* are much simpler, and here is what they look like in the editor:

Type	UserGroup
Parent	None
Group	
Primary Key	7
Version	0
Name Property	name
Sort Property	Use default sorting
Version Property	No version number
Created Property	
Updated Property	
Creator Property	
Updator Property	
Format	
Abstract	☐
Enumerable	☑
Hierarchy Property	
Label Lower	user group
Label Upper	User Group
Labels Lower	user groups
Labels Upper	User Groups
Null Value Label (Read-Only)	Use default
Null Value Label (Read-Write)	Use default
Tooltip	
Visibility	Users
Storage Mechanism	Static
Shareable Table	Not shareable with sub-types
Allows Deletion While Referenced	Does not allow deletion
Table Name	user_groups
ID Column Name	id

© 2010-2015 Exprodo Software

Here is a description of what each *meta property* is for:

Name	Data Type	Description
Type	String	This is the internal string that represents the *Biskit's* type.
Parent	BiskitDef	Some *Biskit's* extend others (in the programming subclass sense) and this indicates the *Biskit Type* that is the parent of this one.
Group	String	Only used by the *Bakery* 446 itself, use any group name required on a *BiskitDef*. Then filter the *BiskitDefs* that are displayed by setting a filter *Biskit* group.
Primary Key	Integer	This is the unique database identifier for the *BiskitDef*.
Version	Integer	Used internally to manage changes to the *BiskitDef*. Increased by one every time a change is made.
Name Property	String	The name of a *property* on the *Biskit* that can be used to represent that *Biskit*. This is not used if **Format** is specified. Important to do this or the **Calpendo** chooses the name for each instance of the *Biskit* that is created.
Sort Property	String	The name of the *property* that should be used to sort *Biskits* of this type.
Version Property	String	The name of the *property* that should be used to hold the version number of the *Biskit*. Only those *properties* of type **Int** with an **Automated Property Type** of **Create & Update** will be used.
Created Property	String	The name of the *property* that should be used to hold the created date and time of the *Biskit*. Only those *properties* of type **Datetime** with an **Automated Property Type** of **Create** or **Create & Update** will be used.
Updated Property	String	The name of the *property* that should be used to hold the last modified date and time of the *Biskit*. Only those *properties* of type **Datetime** with an **Automated Property Type** of **Update** or **Create & Update** will be used.
Creator Property	String	The name of the *property* that should be used to hold the creator of the *Biskit*. Only those *properties* of type **Biskit** with **Biskit Def** set to **User,** with an **Automated Property Type** of **Create** or **Create & Update** will be used.
Updator Property	String	The name of the *property* that should be used to hold the updator of the *Biskit*. Only those *properties* of type **Biskit** with **Biskit Def** set to **User,** with an **Automated Property Type** of **Update** or **Create & Update** will be used.
Format	String	Lets the user specify a format for representing the *Biskit* when there isn't a single *property* that contains the name. See below 356 for more details.
Abstract	Boolean	True if there can be no instances of this *Biskit Type*, only instances of subtypes.
Enumerable	Boolean	True if there are likely to be a large number of *Biskits* of this type, too many for display in a list.

© 2010-2015 Exprodo Software

Hierarchy Property	String (property name)	Some *Biskits* represent a hierarchy. For example, there might be a **Location** *Biskit,* where the **Location** "Oxford" is a child of the **Location** "England", which is a child of "United Kingdom" which is a child of "Europe". A hierarchy is a *Biskit* that has a *property* containing a *Biskit* of the same type, representing the parent in the hierarchy, and also a *property* containing a set of its children. So, Europe has children "United Kingdom", "France", "Germany" and so on. When a user selects a *Biskit* from a hierarchy, they are presented with a "tree" widget making it easier for them to choose the right item. For this to work, **Calpendo** needs to know which *property* contains the children. This must be a **OneToMany** *Biskit property*.
Label Lower	String	Text used to display the *Biskit Type*, when the context calls for a lower case label.
Label Upper	String	Text used to display the *Biskit Type*, when the context calls for an upper case label.
Labels Lower	String	Text used to display the *Biskit Type*, when the context calls for a plural lower case label.
Labels Upper	String	Text used to display the *Biskit Type*, when the context calls for a plural upper case label.
Null Value Label (Read Only)	String	The label to show for *Biskits* of the type represented by this *BiskitDef* when that *Biskit* is null, and we're displaying in a read only context.
Null Value Label (Read Write)	String	The label to show for *Biskits* of the type represented by this *BiskitDef* when that *Biskit* is null, and we're displaying in a read write context.
Tooltip	String	Text to display as a tool tip describing this type of *Biskit*.
Visibility	Nobody, Root, Admins, Users or Everybody	Indicates who can see that this *Biskit Type* exists.
Storage Mechanism	Static or Dynamic	Indicates whether this *BiskitDef* is defined statically, in the source code for **Calpendo**, or dynamically in the *Bakery*.
Shareable Table	Boolean	**Shareable with sub-types** if you want dynamic sub-types to be able to share this *BiskitDefs* database table.
Allows Deletion While Referenced	Boolean	**Allows Deletion** if you want *Biskits* of this type to be deletable even while something references them.
Table Name	String	The name of the database table for storing instances of this *BiskitDef*. The table name for a static *BiskitDef* cannot be modified, since it is defined in the source code for **Calpendo**. Also, not all static *Biskit Types* will have a table associated with them, for example if they are *abstract* or if they are only ever used as components of another *BiskitDef*.

© 2010-2015 Exprodo Software

ID Column Name	String	The name of the column which provides the primary key into the table. In a **Master-Slave** combination these must be different in the **Master** and each **Slave**.

Biskit Format

Set the **Name** *property meta-property* of a *Biskit* to specify the name of the *property* that should be used to represent the *Biskit*. when something more complex than just the label is required. For example, a user has a login name, a given name and a family name, but any one alone can not be used to display the person's name. Instead, a combination of them is required.

Biskit Format Definition

The syntax of the format is rather convoluted, and there will soon be an editor that will let you configure it without having to understand the syntax.

Dynamic Biskits

Use the *Bakery* to create new *Biskit* definitions. These will be *dynamic Biskits*. The user can use **Calpendo** to handle create, read, update and delete values in any table with the following restrictions:

- There must be an integer-valued primary key
- The new dynamic *Biskit* must contain only dynamic *properties*. This means that, if an extra *property* is added to the dynamic *Biskit*, the database table must contain a column for it. If the column doesn't already exist, then **Calpendo** can create the column for the user, or the user can create it.

Dynamic *Biskits* can be used in exactly the same way as **static** *Biskits*.

Creating Biskits

Biskits can have links to each other via *properties*, also some *meta-property* values of a *Biskit* depend on having appropriate *properties*, which means that creating *Biskits* and their *properties* is not a linear event. i.e. the user cannot start at the beginning and hope to move step by step to the end saving completed *Biskits* as they go. Sometimes the user will need to create a *Biskit* with its *properties*, save it with errors, create a second *Biskit*, save it with errors, go back to the first *Biskit*, set up new information now available in the first *Biskit* because the second *Biskit* has been created, re save the first *Biskit*, go back to the second *Biskit*, update that with information from the first *Biskit* and save. This means when creating your own *Biskit's* make sure there is a plan on how they are going to connect together.

© 2010-2015 Exprodo Software

Inheritance

Biskits can inherit from other *Biskits*, inheriting all their *properties* To do this set the **Parent** *meta-property* to be the *Biskit* the new *Biskit* is inheriting from.

When a *Biskit* inherits from another *Biskit* it needs to be specified whether both *Biskits* will share the same table or whether the new *Biskit* will have a table of its own for its own *properties* (those it does not share with the parent). This is done by setting the **Share Super Type Table** *meta-property*. A *Biskit* can only share the table of the *Biskit* it inherits from, if that *Biskit* has the **Shareable Table** *meta-property* set to allow it. When inheriting from the **Booking** *Biskit Type* make sure that the parent and child share a table in order to be able to mutate *booking* from one type to the other effectively, (this is the default).

This may seem very similar to copying a *Biskit*, but there are certain major differences:

- When editing the child *Biskit* the parents *properties* are not seen.
- After adding a *property* to the parent the child automatically has access to the new *property*.
- When changing any *meta-properties* of *properties* in the parent the child *Biskit* will automatically inherit the changes. (e.g. Default Value).
- A *property* in the child cannot be created using the same name as a *property* in the parent.
- If the child *Biskit* has a sibling (same parent *Biskit*) and they all share the same table, do not create *properties* of the same name in both child and sibling. (Currently there are no checks for this but it will cause problems in the database).
- Automatic emails applied to a super-type will automatically apply to any sub-type.
- *Permissions* 448 applied to a super-type will automatically apply to any sub-type but may be overridden in the sub-type.
- If there is a booking sub-type defined, then:
  1) **Booking Rules** 224 will let you choose the sub-type they apply to. A *Booking Rule* 446 that applies to a super-type will also apply to all descendant types too.
  2) A *Booking Rule* that only applies to a particular subtype will only run on a *booking* that both is and was of that subtype or a descendant thereof. This means that if a booking is mutated from one subtype to another, if must be of a suitable type both before and after mutation for the *Booking Rule* to run.
  3) *Bookings* can be mutated from one **Booking** super/sub type to another. Any *property* information not shared by the two **Booking** *Biskit Types* will be lost.

© 2010-2015 Exprodo Software

6.17.3 Property Definitions

A **PropertyDef** is the definition of a *property*[448] on a *Biskit*[446]. In the example of a **Session**, described in Biskit Definitions[353], we saw that a **Session** has 6 *properties*. Therefore, the *BiskitDef*[446] describing a **Session** contains 6 **PropertyDefs**.

A **PropertyDef** itself has *properties*. For example, it needs to record the name and type of the *Biskit property* it represents. There are quite a few different *property* types:

Type	Description	Type-Specific Content
Biskit	*Properties* of type *Biskit* contain a *reference*[449] to another *Biskit*.	See Biskit Property Definitions[363].
Boolean	Contains a true/false value	See Boolean Property Definitions[364]
Date	Contains a date without a time element	None
DateRange	Contains a range of date/time values	None
Datetime	Contains a date and time	None
Double	A floating point number	See Double Property Definitions[365]
Int	The *property* contains a 32-bit signed integer. This may be interpreted via a drop-down selector if the **PropertyDef** indicates this is a Mapped Integer[366], and may allow you to set multiple values if it is a Bit Set[368].	See Integer Property Definitions[366].
JavaEnum	The *property* contains a value that is from an enumeration defined in the source code. This is represented by a drop-down list.	See Java Enum Property Definitions[369]. Must specify the Java Enum Definition[369].
Long	The *property* contains a 64-bit signed integer. Javascript does not directly support 64 bit values, and so long *properties* may cause performance problems because **Calpendo** has to emulate the behaviour of a 64 bit integer in the browser.	None
Set	The *property* contains a set of items. The **PropertyDef** also needs to define the type of the data contained in the set.	See Set Property Definitions[370].
String	A text *property* value. The value may be constrained, by specifying a Mapped String[373], in which case the value would be edited using a drop-down list. Otherwise, it will be a free text entry.	See String Property Definitions[372].
StringEnum	A *property* that contains a string constrained by a String Enumeration[375]. Note that these *properties* are deprecated and may be removed in a future version of **Calpendo**.	See String Enum Property Definitions[374]. Must specify the String Enumeration[375]
UserDefined	Used when the actual type isn't known in advance. Normally used only for indirect *properties*.	None

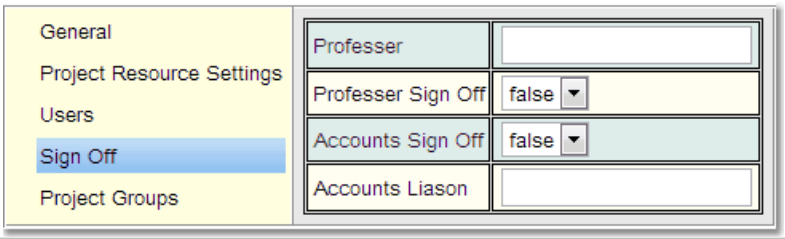
© 2010-2015 Exprodo Software

The different *property* types have different requirements about what information must be stored. For example, an integer *property* needs to specify whether it is to have values constrained by being a Mapped Integer[366] or a Bit Set[368], and in both cases in needs to know which particular Mapped Integer[366] should be used. But other *property* types do not need to know anything about *Mapped Integers*[447].

The requirements of the individual *property* types are shown in the following sections, the rest of this section will focus on the *properties* they have in common.

Primary Key	0
Type	String
Name	myProperty
Description	☑ None
Label	My Property
Tooltip	☑ None
Sort Order	25
Live	false
Min	☑ None
Max	☑ None
Attributes	☑ Visible ☑ Editable ☑ Persistent ☑ Null Allowed ☐ Required ☑ Visible In Biskit Detail ☑ Visible In Biskit List ☐ Visible In Collection Editor
Rows	☑ None
Columns	☑ None
Group	☑ None
Storage Mechanism	Dynamic
Formulaic	☐
Column Name	my_property
Unique	☐
Automated Property Type	None
String Property Type	Single Line
String Type	Unconstrained
Default Value	Null

© 2010-2015 Exprodo Software

Name	Description
Primary Key	This is the database ID for the **PropertyDef**. This cannot be changed, it is only for information purposes.
Type	The type of the *property*, integer, long, date etc
Name	The name of the *property*. This must be unique amongst all the *property* definitions on its *Biskit*, and contain only letters, numbers and the underscore. In particular, this cannot contain spaces.
Description	A description of the content of this property. This is only used for display in the *Bakery* 446 so that anybody using the *Bakery* will be able to see this text. It's not used anywhere else.
Label	This is the text used to display the *property* name. This is used whenever a *Biskit's* detail is displayed, with each *property's* label show before its value.
Tooltip	This is a tool tip that should be displayed whenever the mouse hovers over this *property* when it is being displayed or edited in a *Biskit*.
Sort Order	Used to change the order in which *properties* are displayed. This cannot be modified directly. Instead, *property* sort order is modified by drag-and-drop in the *Bakery*.
Live	Used internally.
Min	The minimum value the *property* can take. This is only used for numeric and string values. For numeric properties, the minimum should be a number representing the minimum value the *property* should have. For string *properties*, the minimum should be the minimum length of the string. If a minimum is set the string is automatically required.
Max	Similar to **Min**, this specifies the maximum value the *property* can take and is only used for numeric and string values.
Attributes	These are defined in a separate table, below.
Rows	The number of rows used to display the value. Particularly useful for multi-line string values.
Columns	The number of columns used to display the value. Particularly useful for multi-line string values.
Group	This can be any text required. In some contexts, *Biskits* are displayed with their *properties* in separate groups. In that case, it's the **PropertyDef's** *Group* string that defines the group that each *property* belongs to, and the text that should be used to name that group. If this is used, when viewing a *Biskit,* separate tabs each with the appropriate group name will be used to display the properties. This *property* is used if no layout has been defined in the layout editor for this *Biskit Type* 446. The **Project** *Biskit* will always use this *property* for its layout. 

© 2010-2015 Exprodo Software

Storage Mechanism	Indicates whether the *property* is static, dynamic or indirect. See Property Storage Mechanisms 351 for more information.
Formulaic	Dynamic only. Allows a formula to be entered to calculate the value of this *property*. See the section on Formulae 375 for more details on the formatting.
Column Name	Only required for dynamic *properties*, this is the name of the database column used to hold the *property*.
Unique	Only required for dynamic *properties*, this indicates whether the value stored in this *property* should be unique. If it is, then when **Calpendo** is asked to generate SQL code to update the database schema, it will add the necessary commands to tell the database that the column contains unique values. This will make it illegal for the same non-null value to be present twice.
Automated Property Type	This specifies whether the *property* is automatically assigned a value, and when. The potential values are: **None, Create, Update or Create & Update.** If a property has a value other than **None** then the property will not be viewable when a *Biskit* is being edited.
Default Value	The value the user will see on creation. If this value is outside the **Min** and **Max** values above then the user will have to update before saving.

The *attributes* on a *PropertyDef* are all boolean values as follows:

Name	Description
Visible	Indicates whether the *property* should ever be displayed. If you make a *property* invisible, it will always be hidden.
Editable	Specifies whether this *property* can ever be modified. Some static *properties* are special and cannot be changed, but you may sometimes have dynamic *properties* that are calculated from some other source and inserted into the database. You can mark such *properties* as not being editable to protect them from accidental change.
Persistent	Indicates whether the *property* is stored in the database. Dynamic and indirect *properties* should always be persistent.
Null Allowed	Specifies whether null is an allowed value for the property. See below 362 for a discussion about **Null Allowed** and **Required**.
Required	Specifies whether it is mandatory to provide a value for the property. See below 362 for a discussion about **Required** and **Null Allowed**.
Visible In Biskit Detail	Specifies whether this *property* should be visible when the full detail of a *Biskit* is displayed.
Visible in Biskit List	Specifies whether this *property* should be visible when a list of *Biskits* is displayed.
Visible in Collection Editor	Specifies whether this *property* should be visible when displaying a collection of *Biskits* of this type. Normally, when editing a collection of *Biskits*, the number of *properties* displayed would be kept very small.

© 2010-2015 Exprodo Software

Required Fields And Null Allowed

Required and **Null Allowed** sound like they are exact opposites, and therefore **Calpendo** shouldn't support both of them. However, there is a subtle difference that means that it is sometimes useful to define a *property* that is both required and allows null. The **Null Allowed** attribute is checked when building an editor for a *property*. So, when displaying a drop-down for a *Mapped Integer*, *Mapped String*[447], **Java Enum** or **String Enum**, a value representing **null** will be added if the **Null Allowed** attribute is set to **true**.

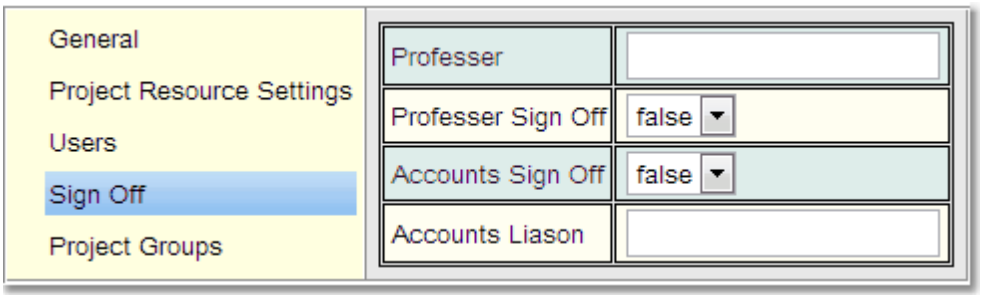
When a new *Biskit* is being created, the default value of a *property* displayed with a drop-down with **Null Allowed** set to **true** will be **null**, since that is the first value shown in the drop-down.

However, if a property is set to **Required**, then a value must be provided; it cannot be saved without a non-null value. For properties that have **Required** and **Null Allowed** both set to **true**, this means that the user is forced to choose one of the values from the drop-down as the default value of null cannot be left as the current value. A drop-down for a *property* that is **Required** but for which **Null Allowed** is **false** would be built without a value representing **null**, and so the user would not be forced to choose. Instead, the first value in the drop-down would be automatically selected unless a choice is made by the user.

For string properties, an empty string is a non-null value and so will satisfy **Required**. In order to make sure the user can not do this set a minimum length for strings where you want to make sure they enter something.

Large Numbers Of Properties And Tab Layout

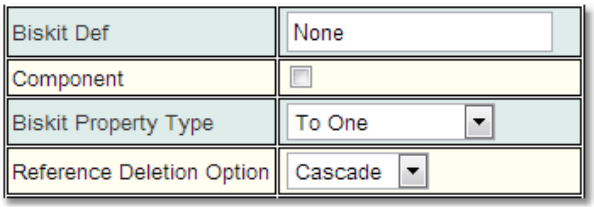
If you are adding a large number of *properties* to a *Biskit* and wish them to be displayed in particular groups under specific tab names then use the **Group** *meta-property*. All *properties* with the same **Group** *meta-property* will be displayed under the same tab which will be labelled with the value of the **Group** *meta-property* unless a layout has been defined for this *Biskit* in the **Layout Editor**. (The **Project** *Biskit* will always use **Group** for its layout)



© 2010-2015 Exprodo Software

6.17.3.1 Biskit Property Definitions

A *property*[448] of type *Biskit*[446] must specify the following information:



Name	Meaning
BiskitDef	Specifies the type of *Biskit* this *property* will contain.
Component	Indicates whether the *Biskit* value is stored as a *reference*[449] to a *Biskit* stored somewhere else, or if its *properties* are stored directly as a component of its owning *Biskit*. Only used for static *properties*.
Biskit Property Type	**To One**, **Many To One**, **Master to Slave** or **Slave To Master** explained below[363].
Inverse Property	Must be provided when this is a **Many To One** property, **Master to Slave** or **Slave to Master**, in which case this is the name of the *property* on the parent that points to the child *BiskitDef*[446].
Reference Deletion Option	Whether this objects of this *Biskit* type can be deleted if *referenced* by other *Biskits* in the DB. **Cascade**, **No Action**, **Set null**

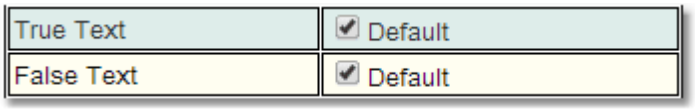
Biskit Property Type

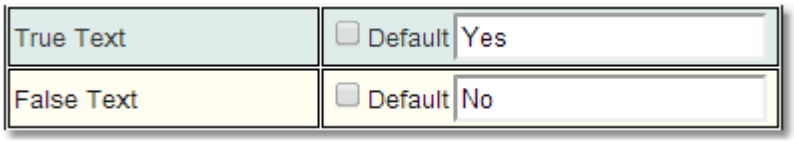
Value	Meaning
To One	The value of this *property* will *reference* a *Biskit*, but that *Biskit* will not know anything about us *referencing* them.
Many To One	The value of this *property* will *reference* a *Biskit* in a child-parent relationship. We are the child, and this *property references* the parent. The parent will also have a **Set** or **List** *property* that contains its child *Biskits* using a **One To Many** *property*. With this value of **Biskit Property Type**, we must also specify the **Inverse Property** that is the name of the parent's *property* that contains the children.
Master to Slave	When a *Biskit* is required which has a large number of *properties*, ie. in excess of a thousand in SQL, the DB cannot cope with this. In order to do this create **Slave** *Biskit's*, that is *Biskits* that are created, deleted and *referenced* with their **Master**. The **Master** *Biskit* would need a *property* of type *Biskit* which would point to each **Slave** and be a **Master to Slave** type, the **Slave** would need a *property* of type *Biskit* which would point to the **Master** and be a **Slave to Master** type. A **Master** can have many **Slaves** in order to reference as many *properties* as needed. The only caveat is that the **Primary Key Column Name** for any **Slave** *Biskit* must be different from the **Masters** (defaults to **id**, therefore needs to be changed), and different from any other **Slave** of that **Master**. The **Inverse Property** also needs to be set up in both *BiskitDefs*. See Creating a Master-Slave Biskit Relationship[407] example.
Slave To Master	

© 2010-2015 Exprodo Software

6.17.3.2 Boolean Property Definitions

A boolean *property*, as well as the normal settings required by a **PropertyDef**, may also specify the text to be displayed instead of **True** and **False**.





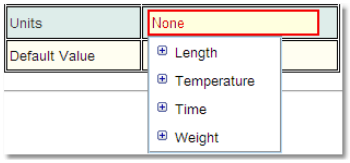
By un-ticking the **Default** option the user can then specify which text is to be used to display the value instead of **True**/**False**.

When used in a formula the values are **T** or **F.**

© 2010-2015 Exprodo Software

6.17.3.3 Double Property Definitions

A double *property*, as well as the normal settings required by a **PropertyDef**, may also specify its **Units**. Doubles are numbers with a decimal point.



There are four types of units **Length**, **Temperature**, **Time**, **Weight**. Users can also add their own units into **Calpendo**, Each type of unit has a number of options, and these are:

Name: Length

Units

Name	Short Name	Multiplier	Sub Units Per Unit	Offset	Sub Unit
Inches	in	0.0254	10	0	
Yards	yd	0.9144	10	0	
Miles	miles	1609.344	10	0	
Feet	ft	0.3048	12	0	Inches
Metres	m	1	10	0	
Millimetres	mm	0.001	10	0	
Centimetres	cm	0.01	10	0	
Feet (decimal)	ft	0.3048	10	0	

Name: Temperature

Units

Name	Short Name	Multiplier	Sub Units Per Unit	Offset	Sub Unit
Centigrade	C	1	10	0	
Kelvin	K	1	10	-273.15	
Fahrenheit	F	0.555555555555556	10	-17.7777777777778	
Celsius	C	1	10	0	

Name: Time

Units

Name	Short Name	Multiplier	Sub Units Per Unit	Offset	Sub Unit
Seconds	s	1	10	0	
Minutes	mins	60	10	0	
Hours	hours	3600	10	0	
Days	days	86400	10	0	
Micro Seconds	us	0.000001	10	0	
Milli Seconds	ms	0.001	10	0	
Weeks	weeks	604800	10	0	
Months	months	2629800	10	0	
Years	years	31557600	10	0	

Name: Weight

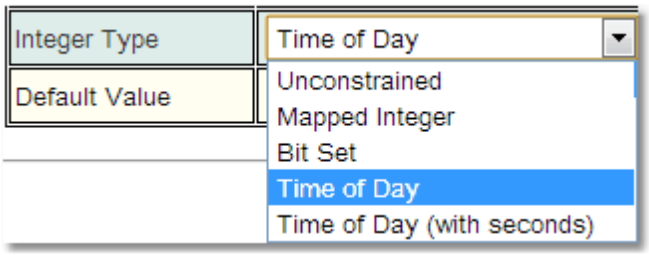
Units

Name	Short Name	Multiplier	Sub Units Per Unit	Offset	Sub Unit
Stones	st	6.35029318	14	0	Pounds
Kilograms	kg	1	10	0	
Grams	g	0.001	10	0	
Ounces	oz	0.028349523125	10	0	
Pounds	lb	0.45359237	10	0	
Stones (decimal)	st	6.35029318	10	0	

© 2010-2015 Exprodo Software

6.17.3.4 Int Property Definitions

An integer (a whole number, no decimal information) *property*, as well as the normal settings required by a **PropertyDef**, must also specify its **Integer Type**. This indicates whether the content of the integer value should be constrained. The values allowed are:

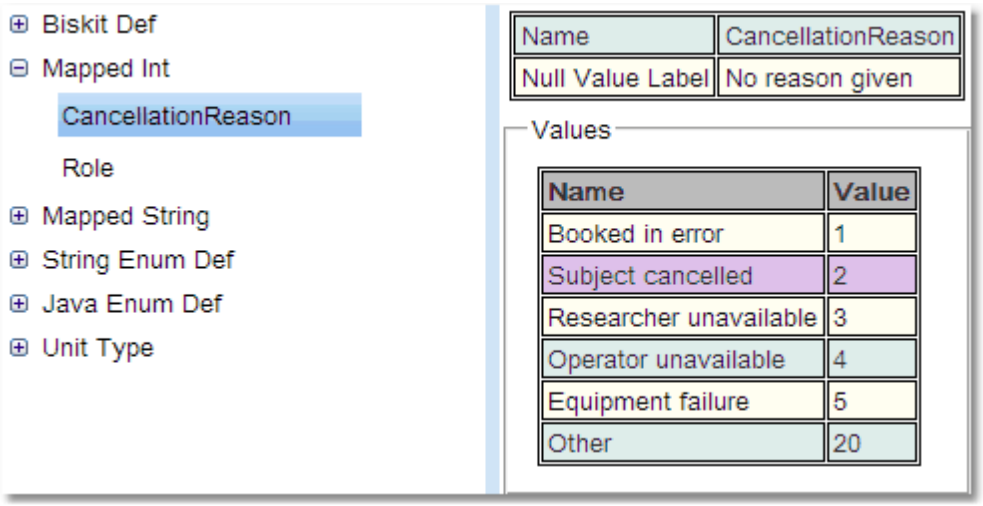


Value	Meaning
Unconstrained	The value can be anything.
Mapped Integer	The value will be one of the values specified by a Mapped Integer[366]. The particular *Mapped Integer*[447] that constrains the value must be specified.
Bit Set	The value will be a Bit Set[368] containing values specified by a Mapped Integer[366]. The particular *Mapped Integer* that constrains the value must be specified.
Time of Day	Stored in the database as the number of seconds since midnight. The only difference is the formatting on output, with or without seconds.
Time of Day (with seconds)	

6.17.3.4.1 Mapped Integers

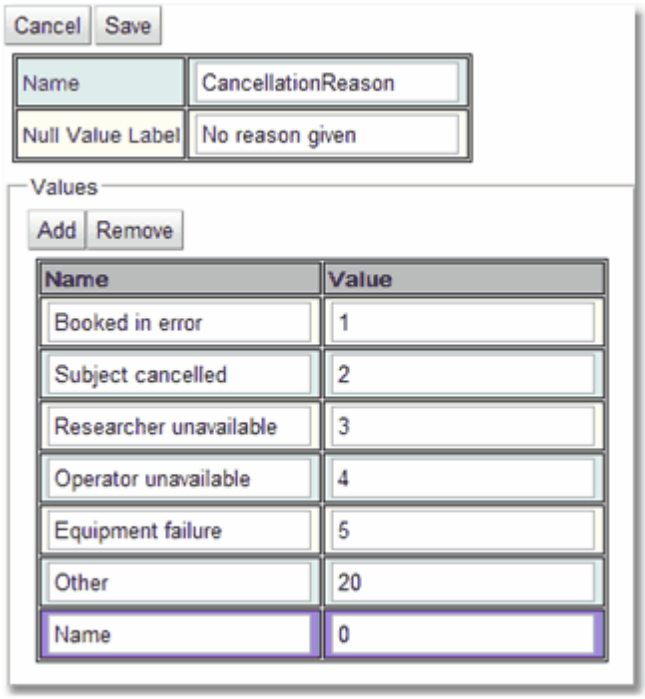
A *Mapped Integer*[447] is a mapping between a text display value and an integer value. It provides a way for a user to edit an integer *property* by seeing a drop-down list of text values, hiding the numerical values from them. This provides a means of generating drop-down boxes for entering values where there is a fixed set of valid values that can be entered, while storing an integer in the database.

Here is an example of a *Mapped Integer* defined in the database:

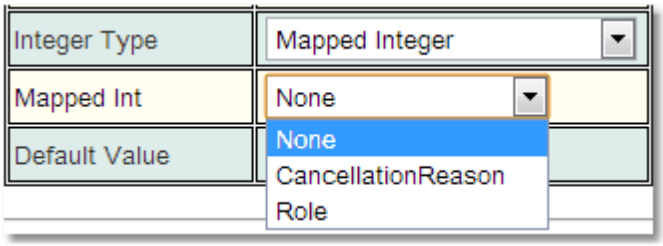


© 2010-2015 Exprodo Software

To add new values press the **Edit** button and then the **Add** button, the new entry will appear at the bottom, set the text string to be viewed and the number to store it as. Then press **Save**.



There can be as many *Mapped Integers* as needed in the database. Once the *Mapped Integer* is created, to use it, create an integer *property* on a *BiskitDef*[446] and set its **Integer Type** to be *Mapped Integer* and chose the *Mapped Integer* you have created.



The database stores the values as integers. The definition of the *Mapped Integer* in the *Bakery*[446] just maps those numbers to a string for display. At a later date new items can be added to the mapping (new strings and new integers), the mapping can be changed by changing the text associated with a particular integer, then everything in the database that has that integer value will be displayed with the new string.

If there is a mapping where 1 displays as "A", and 2 displays as "B", and then this mapping is changed so that 2 displays as B and 3 displays as A, and there is no mapping for 1, then there won't be a way to display a text value for 1, and there won't be a way to search for it.

© 2010-2015 Exprodo Software

6.17.3.4.1.1 Bit Sets

A *Bit Set*[446] is a *BiskitDef*[446] integer property that uses a special sort of *Mapped Integer*[447]. To be used as a *Bit Set*, a *Mapped Integer* must map integer values that are between 0 and 31. It doesn't need to map all 32 possible values, but the values mapped must be within that range.

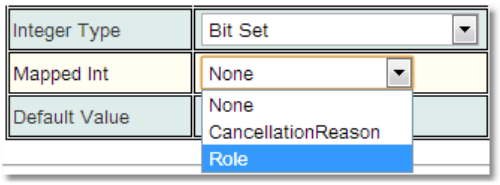
A *Bit Set* works by using binary values. Each *Bit Set* has a particular value, and when a number of bits have been set the value of the integer is the total of each of the individual bits values added up.

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	.....
Value	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	.....

Therefore if bits 1, 5 and 7 are set, the value is 162. (2+32+128)

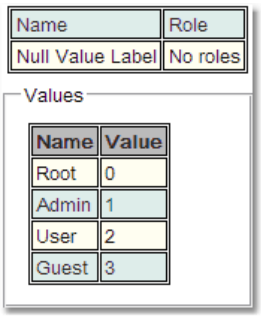
Bit Sets allow the user to combine up to 32 options in any combination required. In **Calpendo** we have one default bit set and that is for User Roles[368].

There can be as many *Bit Sets* as needed in the database. Once the *Bit Set* is created as a *Mapped Integer*, to use it, create an integer *property*[448] on a *BiskitDef* and set its **Integer Type** to be *Bit Set* and chose the *Mapped Integer* that has been created.



6.17.3.4.1.2 User Roles

There is a special *Mapped Integer*[447] definition called *Role*[450]. This is used by the *Roles* bit set property of a **User**. Since it is used as a *Bit Set*, it may only have values of 0 to 31. Value zero is special in this *Mapped Integer*. Any user whose roles *property*[448] has bit 0 set is a **Root** user and Permissions[278] do not apply to such a user.



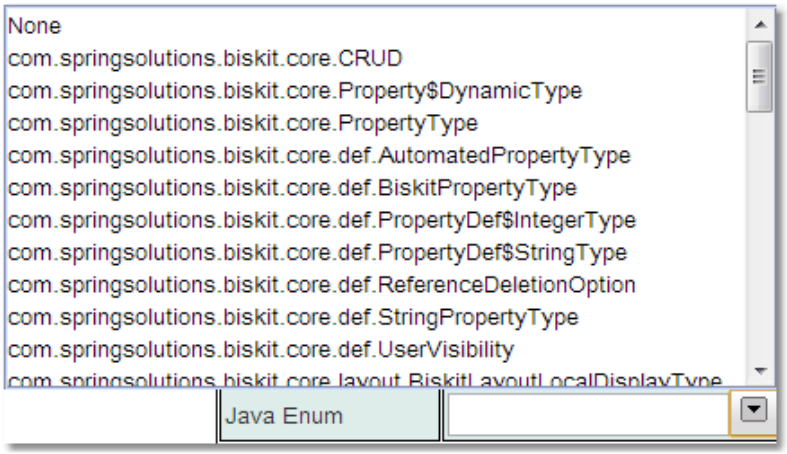
So using the Bit Set[368] bit table a user that has both **Admin** and **User** roles would have a value of 6, the 1 and 2 bits set. A **Guest** user would have the value 8, just the 3 bit set.

The user may add additional values to this *Mapped Integer*, and may rename any of them (including **Root**)

© 2010-2015 Exprodo Software

6.17.3.5 Java Enum Property Definitions

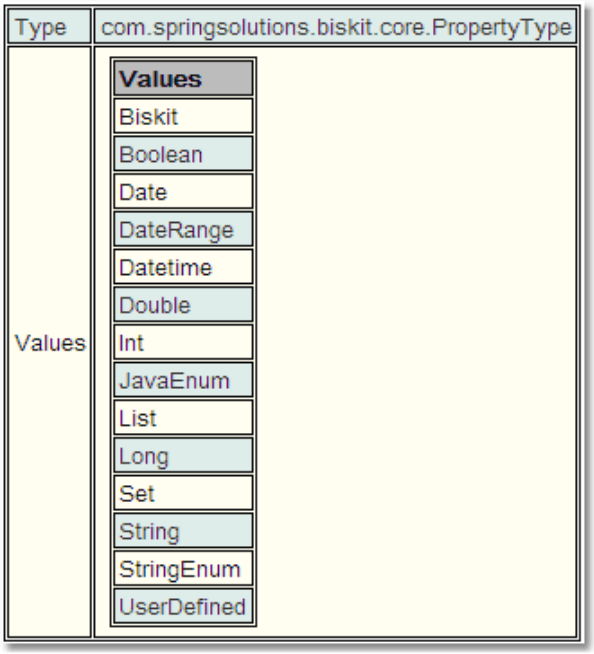
A **Java Enum** *property*[448], as well as the normal settings required by a **PropertyDef**, must also specify its **Java Enum**. This indicates which of the **Java Enums** to be used.



6.17.3.5.1 Java Enum Definitions

A **Java Enum** is a representation of an enumeration defined in the original Java source code. End users should not create, change or delete any **Java Enum** defined in **Calpendo**. You may create *properties*[448] of type **Java Enum** and use any of the existing enumerations, but you should do so carefully in case the enumeration changes or is deleted in a future release.

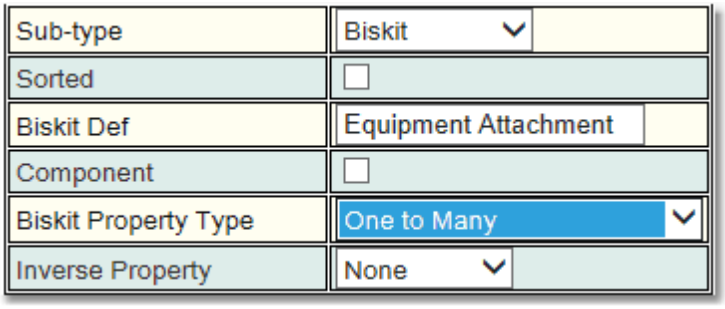
Properties of type **Java Enum** are edited with a drop-down list box showing the values from the enumeration. In special cases, such a drop-down list box might display only some of the values from the enumeration. However, it is not possible for a user-defined *property* to behave in this manner.



© 2010-2015 Exprodo Software

6.17.3.6 Set Property Definitions

A *property*[448] of type **Set** must specify the *Property Type*[448] of the elements within the **Set**. If the elements are not *Biskits*[446], then there are no further settings required. If the elements are *Biskits*, then the following must be specified:



Name	Meaning
Sub-type	The type of data stored by the *property* represented by this definition. For types other than **Biskit** the only additional *meta property*[448] will be **Sorted**.
Sorted	Indicates whether the contents of the **Set** are sorted or not.
BiskitDef	Specifies the type of *Biskit* the set will contain.
Component	Indicates whether the *Biskit* value is stored as a *reference*[449] to a *Biskit* stored somewhere else, or if its *properties* are stored directly as a component of its owning *Biskit*. Normally only used for static *properties*.
Biskit Property Type[371]	**One to Many**, **Many to Many Persisted**, **Many To Many Unpersisted**, **To Many**, explained below[371].
Inverse Property	This is the property on the **Set** *Biskit* which will point back to the parent *Biskit*

© 2010-2015 Exprodo Software

Biskit Property Type

Value	Meaning
One To Many	Indicates the set contains "child" *Biskits* that will all *reference* this *Biskit* as the "parent" in a **Many To One** *Biskit property*. With this value of *Biskit Property Type*, we must also specify the **Inverse Property Name** that is the name of the child's *property* that *reference* the parent.
Many To Many Persisted	This **Set** represents the persisted half of a **Many To Many** *property*. See Many To Many Properties 370.
Many To Many Unpersisted	This **Set** represents the unpersisted half of a **Many To Many** *property*. See Many To Many Properties 370.
To Many	This **Set** is a collection of *Biskits*, but the *Biskits* we contain do not *reference* us in any way.

Currently **Many to Many** (in all its forms) is not implemented for dynamic **Set** *properties*.

Many To Many Properties

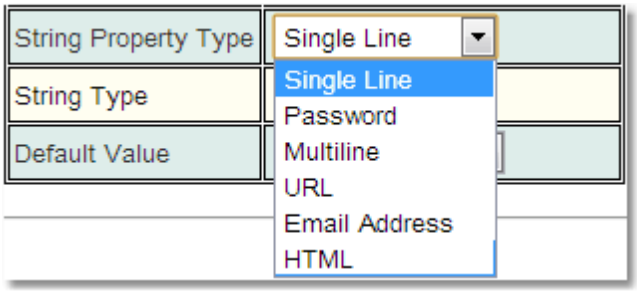
Suppose a user can be associated with any number of *projects* 448, and a *project* can have any number of users associated with it. This means there will be a collection of *projects* stored in a *property* on each user, and also a collection of users stored in a *property* on each *project*. Both sides of this relationship are known as **Many To Many** properties.

If a *project* is a member of a user's *projects*, then that same user would be a member of that same *project's* users and vice versa. This means it is not necessary to save both sides of this relationship to the database because of this symmetry. However, there are some occasions when it is necessary to know which side is persisted, and so one half is known as a **Many To Many Persisted**, and the other half is known as **Many To Many Unpersisted**.

© 2010-2015 Exprodo Software

6.17.3.7 String Property Definitions

Property[448] definitions of type **String** must specify the following information:



Name	Meaning
String Property Type[372]	Defines the nature of the content of the string.
String Type[373]	Indicates whether the content of the text value should be constrained.

String Property Type

This defines the nature of the content of the string, with possible values shown in this table:

Value	Meaning
Single Line	One line of text.
Password	A single line content that should be hidden when typed. Intended for passwords, but can be used for any text that should not be seen as it is typed.
Multiline	A text area with multiple lines. Typically, the **PropertyDef** would also specify the number of rows to display.
URL	A single line of text that is expected to contain the web address of some page. When rendered in a read-only format, it displays with an HTML anchor element so that it will operate as a hyperlink. The content can contain a prefix of the form "http://" or "https://" or any other valid protocol recognised by all web browsers, but the protocol prefix is not required.
Email Address	A single line of text that is expected to contain an email address. When rendered in a read-only context, this is displayed with a "mailto:" HTML link so that it will automatically create an email when clicked.
HTML	A text area that contains HTML. When edited, this *property* will use an HTML editor, with tools for helping you to create valid HTML.
Colour	Three hexadecimal numbers that define the RGB colour to be displayed.

© 2010-2015 Exprodo Software

String Type

This indicates whether the content of the text value should be constrained. The values allowed are:

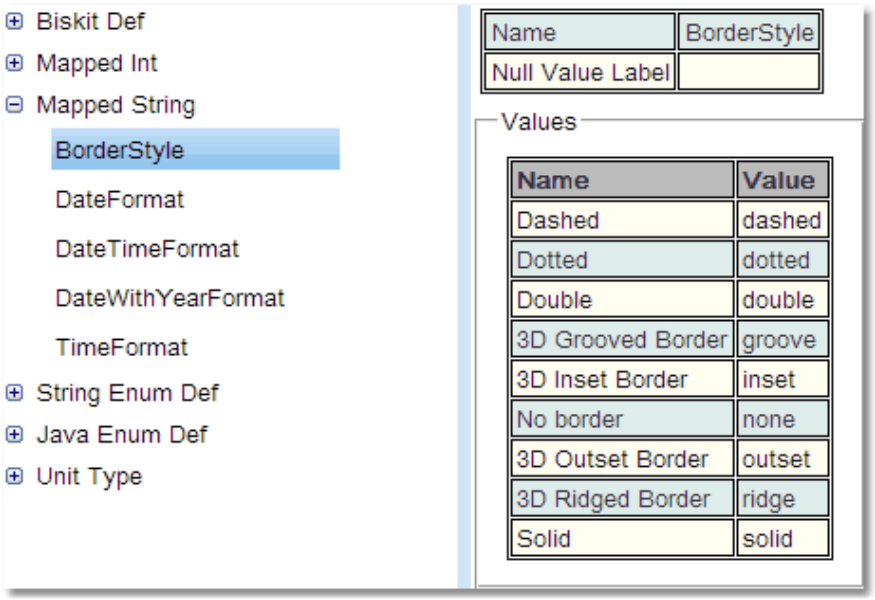
Value	Meaning
Unconstrained	The text value can be anything.
Mapped String	The value will be one of the values specified by a Mapped String[373]. The particular *Mapped String*[447] that constrains the value must be specified.
Biskit Type	The value will be the type of a *BiskitDef*[446], and so will be configured as a drop-down listing all the known *Biskit Types.*[446]

6.17.3.7.1 Mapped Strings

A *Mapped String*[447] is similar in concept to a *Mapped Integer*[447], but instead of storing an integer in the database, a text value is stored instead. This means that a *Mapped String* is a mapping from one set of strings to another set of strings. When a **String** *property*[448] is created on a *BiskitDef*[446], you can indicate that it should be constrained to contain values from a *Mapped String*. Then, when a user edits such a *property*, they will see a drop-down that lists the labels defined in the *Mapped String*, although the value stored in the database will be the associated text value.

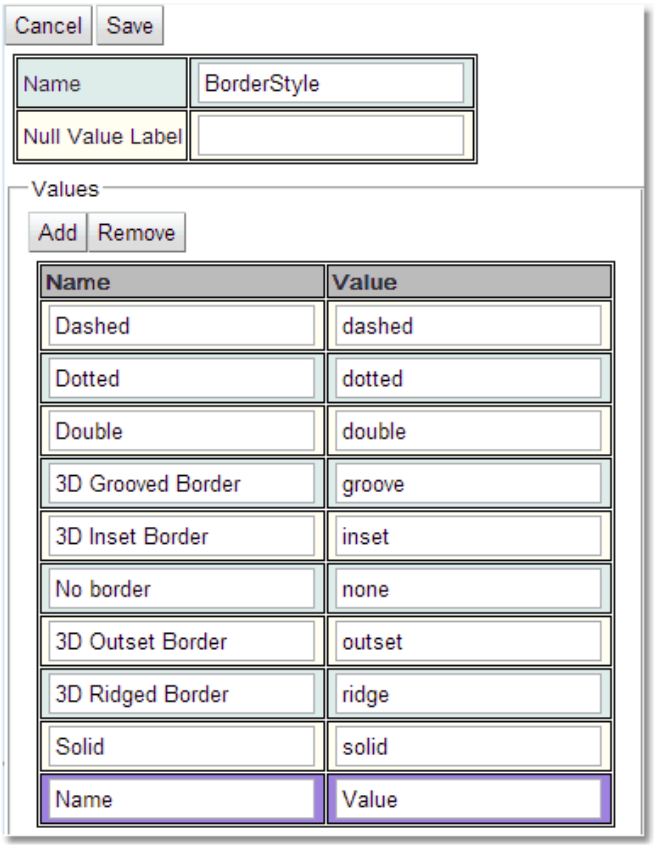
Mapped Strings are similar to String Enumerations[375], in that they both store values as text, and they both generate a drop-down selection list. *Mapped Strings* are the preferred mechanism though, because by decoupling the displayed labels from the values stored in the database, it makes things very easy it you ever want to change the way values are displayed to users. This is why **String Enumerations** are deprecated, and may be removed in a future version of **Calpendo**.

Here is an example of a *Mapped String* defined in the database:



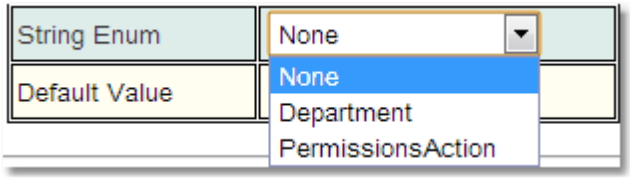
© 2010-2015 Exprodo Software

To add new values press the **Edit** button and then the **Add** button, the new entry will appear at the bottom, set the name of enumeration and the value to store with that name. Then press **Save**.



6.17.3.8 String Enum Property Definitions

A **String Enum** property, as well as the normal settings required by a **PropertyDef**, must also specify its **String Enum**. This indicates which of the **String Enums** to be used.

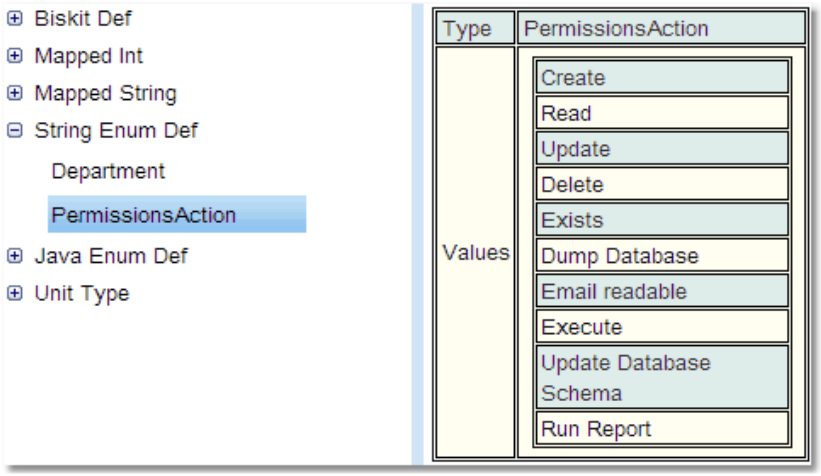


© 2010-2015 Exprodo Software

6.17.3.8.1 String Enumerations

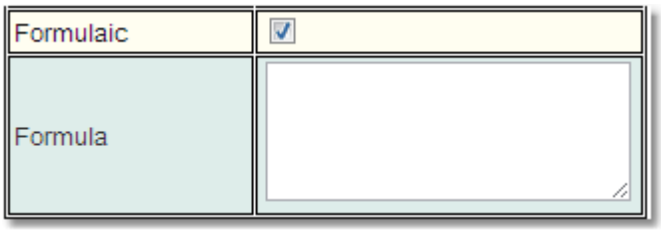
A **String Enumeration** is like a Mapped String[373] in which the name and value are always the same. A *property*[448] with a type of **StringEnum** will be editable with a drop-down list box showing all the values in the enumeration.

String Enumerations are deprecated and *Mapped Strings* should be used instead. In some future release, **String Enumerations** may be removed. This is because, by tightly coupling the displayed text with the text stored in the database, it makes it very difficult to change the text that is displayed.



6.17.4 Formulae

If the **Formulaic** check box is ticked then the value of this *property*[448] will be calculated by a formulae.



The syntax for the formulae is based on MySQL, with one major exception, MySQL uses the column name in its formulae, most users of **Calpendo** would prefer to use the *property* name. In order to do this there is a special syntax, if the *property* name is in [ ], such as [durationInMinutes] then **Calpendo** will convert the *property* name to the correct column name.This also allows the user to specify **[biskit.property]** in order to access *properties* of related *Biskits*[446].

© 2010-2015 Exprodo Software

When creating formulae use all the standard number calculations * / + - , as well as setting up logic to determine the correct answer.

For instance there are two main logical options **if** and **case**:

Logic	Description
if (expr1, expr2, expr3)	If expr1 is true then expr2 will be returned otherwise expr3 is returned. (You cannot use the if statement)
case when expr1 then option1 when expr2 then option2 else option3 end	If expr1 is true do option1, otherwise if expr2 is true do option2, otherwise do option3. Case statements can be nested one inside the other.

An expression can include < > != = () as well as AND or OR for example:

[costPerHour] > 1000 OR (**[costPerHour]** < 500 AND **[costPerHour]** > 275)

When dealing with strings there are a number of functions that can be used, here is a subset of the more useful ones:

Function	Description
length(str)	Return the length of the string
concat(str1,str2)	Return str 1 and str2 concatenated together
replace(str1,str2,str3)	In str1, find where str2 occurs, and replace it with str3
substr(str,pos)	Return the string that starts at the pos character of str.

When dealing with dates there are also many functions some of the more useful are:

Function	Description
datediff(date1, date2)	Returns the difference in whole days between two dates.
unix_timestamp(date)	Returns the number of seconds since midnight 1/1/70.
timediff(date1, date2)	Returns the difference between two times as a time string,
time_to_sec(time)	Returns the conversion of a time string to seconds,

Using **time_to_sec()** in conjunction with **timediff()** allows a calculation of the time in seconds between two times. i.e. **time_to_sec(timediff(date1,date2))**

© 2010-2015 Exprodo Software

Here is an example of a formulae using a nested case and **datediff()**.

In this example the value of a cost *property* of a *booking*[446] is determined depending on the status of a *booking*.
If the *booking* is denied, or cancelled with more than seven days to go then do not charge. If the *booking* goes ahead then charge the rate found in the *property* **costPerHour** in the *Project Resource Settings*[448] for that *project*[448] multiplying by the duration in minutes of the booking divided by 60 to give hours. If the *booking* was cancelled less than two days ago charge 50% otherwise charge 20%.

```
case
  when [status] = 'Denied' then 0
  when [status] = 'Approved' or [status] = 'Requested' then
[projectResourceSettings.costPerHour] * [durationInMinutes] / 60
  else
    case
      when datediff(start_date, [cancelled]) >= 7 then 0
      when datediff(start_date, [cancelled]) >= 2 then
0.2*[projectResourceSettings.costPerHour]  * [durationInMinutes] / 60
      else 0.5*[projectResourceSettings.costPerHour]  * [durationInMinutes] / 60
    end
end
```

Known Formulae Problems

There are a number of issues with using formulae in *Biskits*. As the formulae has to be converted from what is written in **Calpendo** to MySql there are currently some functions that will not work.

1. Any function that requires a string constant as a parameter. These constants do not currently convert from **Calpendo** to MySQL cleanly.
   For example: **timestampdiff**(unit,datetime_expr1,datetime_expr2) where **unit** is one of FRAC_SECOND (microseconds), SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER, or YEAR. **Unit** does not convert properly.

2. The **if** statement does not work so use the **if()** function instead.

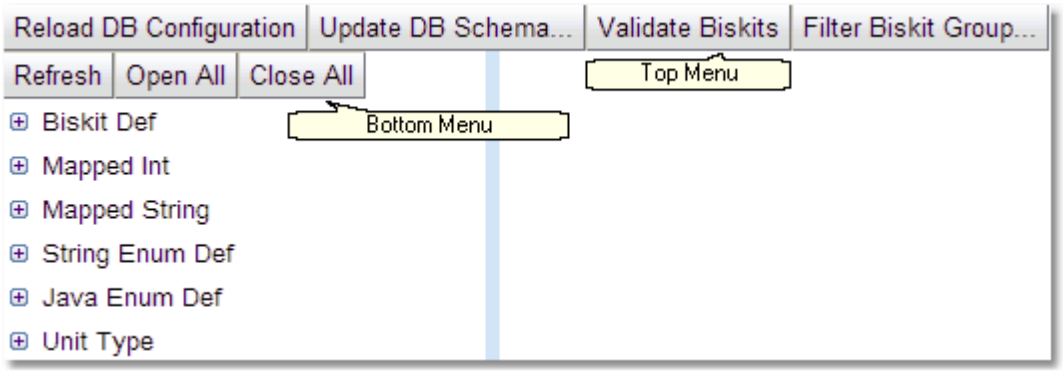
3. Use **'T'** and **'F'** for the value of Boolean properties.

© 2010-2015 Exprodo Software

6.17.5 Bakery Editior

The **Bakery Editor** allows an **Admin** user to see, modify, create and delete *Biskits* [446] and their *properties* [448]. The **Bakery Editor** also allows access to any *Mapped Integers* [447] or *Strings* [447], any **String** and **Java Enumeration** definitions as well as the unit definitions. To access the **Bakery Editor** page it can be found at **Admin->Bakery**. However, the administrator may have configured **Calpendo** so that the menu is different.

When entering the **Bakery Editor**, in the left hand pane will be all the current definitions for *Biskits* and the additional definitions in an expanding tree structure. The right hand pane will hold the information for whatever is selected in the left hand pane.



The top menu provides buttons to check and update the database once the user has completed any changes in the DB.

Top Menu Item	Description
Reload DB Configuration	Tells the server to reconfigure according to the current dynamic *property* definitions you have. Make sure the database schema is up-to-date before running this.
Update DB Schema	Asks the server to generate a SQL script that can be used to create whatever columns are required for the dynamic *property* definitions. After the script has been generated, the user will have the option of applying it.
Validate Biskits	Checks that *Biskit* definitions are self consistent
Filter Biskit Group	Display only those *Biskits* in a particular group. By default no groups are set up, if your own *Biskets* are created they should be put into their own group in order to use this filter to find them quickly.

When changes have been made to the DB via the **Bakery Editor** the user MUST make sure to update the DB properly by using these buttons in the correct order:

1. Press **Update DB Schema** to generate a script to update the SQL database. Then apply the script to make the changes

2. .Press **Validate Biskit** to make sure any *Biskits* that have been changed have a self consistent definition.

3. Press **Reload DB Configuration** to reconfigure the server so that it is using the new DB schema.

© 2010-2015 Exprodo Software

Once this is done the **Calpendo** will now be able to use the changes implemented in the DB through the **Bakery Editor**.

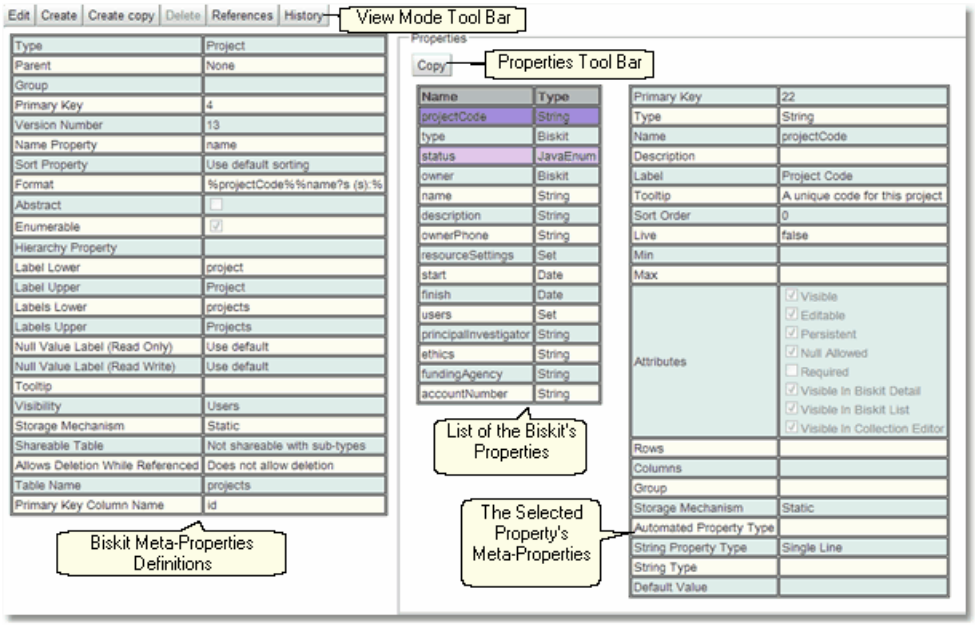
The bottom menu provides buttons to deal with the expanding tree structure.

For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition 96 chapter.

Use these in conjunction with the **Filter Bisket Group** button on the top menu to aid the viewing of the tree information.

View Mode

View mode allows the user to see each of the *Biskits* and their *properties*. You can also copy *properties* for later use in a different *Biskit*.



For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition 96 chapter.

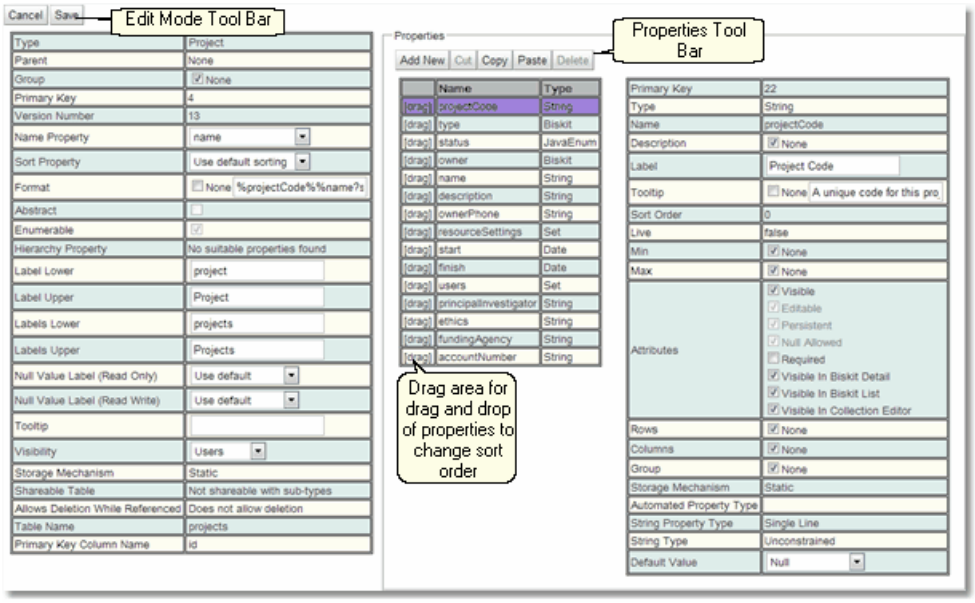
Items that are created will not exist in the database until you press the **Save** button and then follow the procedure to validate, update and reload the DB 378.

Properties Tool Bar	Description
Copy	This button will take a copy of the selected *properties* and place them in a buffer to allow pasting into another *Biskit*. Multiple *properties* may be selected by using **ctrl** and **shift**.

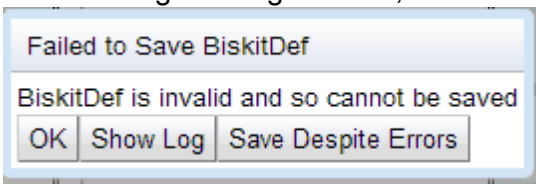
© 2010-2015 Exprodo Software

Edit Mode

Edit mode allows the user to edit a *Biskit* and all its *properties*. Once changes are saved, the DB will need to be validated, updated and reloaded [378], in order for them to be permanent.

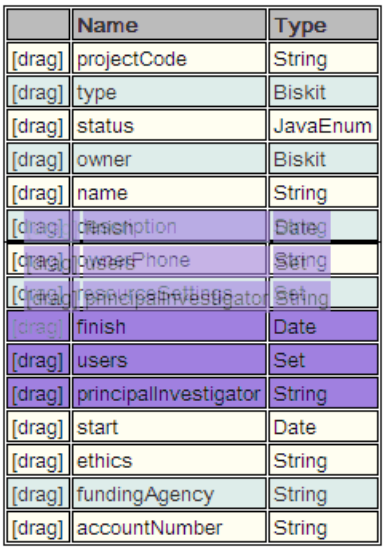


For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition [96] chapter.

Edit Mode Tool Bar	Description
Save	Saves all the changes to the *Biskit* and its *properties*. In order to make these changes permanent follow the procedure to validate, update and reload the DB [378]. When saving a changed *Biskit*, an error message may occur:  Look at the log (**Show Log** button) and then depending on the errors decide whether to **Save Despite Errors**. Sometimes *Biskits* need to be saved with errors due to the need to set up links between *Biskits*, and both *Biskits* need to be created before that can be done. With some errors there is not the choice of whether to **Save Despite Errors**, these errors must be fixed before saving because they can corrupt the database.

© 2010-2015 Exprodo Software

Properties Tool Bar	Description
Add New	Adds a new *property* to the *property* list. The *property* is automatically selected and available for its *meta properties* [448] to be modified.
Cut	Multiple *properties* may be selected by using **ctrl** and **shift**
Copy	Multiple *properties* may be selected by using **ctrl** and **shift**.
Paste	Pastes the *properties* held in the buffer into the current *property* list.
Delete	Multiple *properties* may be selected by using **ctrl** and **shift**.



	Name	Type
[drag]	projectCode	String
[drag]	type	Biskit
[drag]	status	JavaEnum
[drag]	owner	Biskit
[drag]	name	String
[drag]	finish	Date
[drag]	users	Set
[drag]	principalInvestigator	String
[drag]	finish	Date
[drag]	users	Set
[drag]	principalInvestigator	String
[drag]	start	Date
[drag]	ethics	String
[drag]	fundingAgency	String
[drag]	accountNumber	String

Properties are placed in the list of *properties* for a *Biskit* as defined by their **Sort Order** *meta-property*.

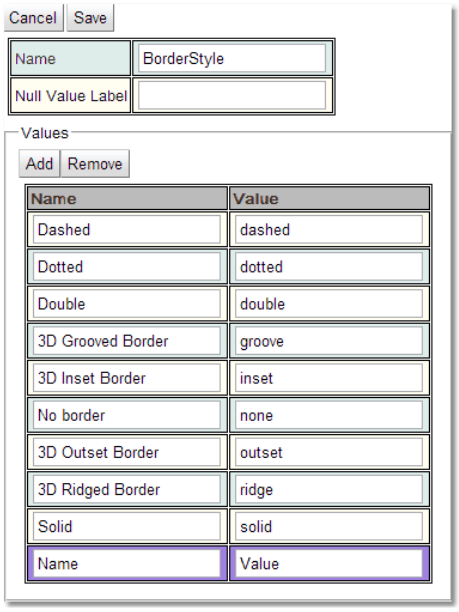
That **Sort Order** *meta-property* cannot be edited directly but changed by using the [drag] area, and dragging and dropping selected properties within the property list.

In the example three *properties* have been selected using the **shift** key and are being dragged up the list to be placed higher up.

© 2010-2015 Exprodo Software

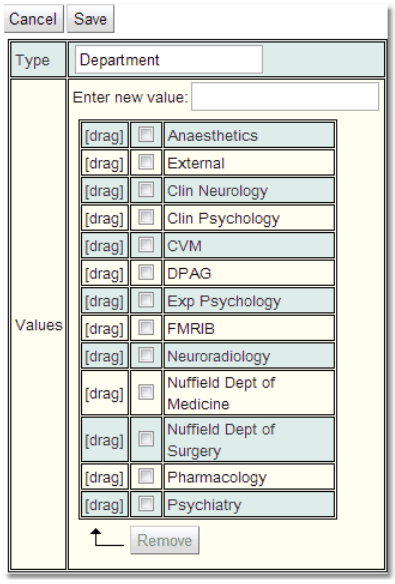
Editing Mapped Int and Mapped String

When in edit mode for *Mapped Int* and *Mapped String* there are buttons to **Add** and **Remove** items. When an item is added, a new **Name Value** pair appear at the bottom of the list to put in the new mapping. To remove items select a **Name Value** pair first then press the **Remove** button.



Editing String Enum Def

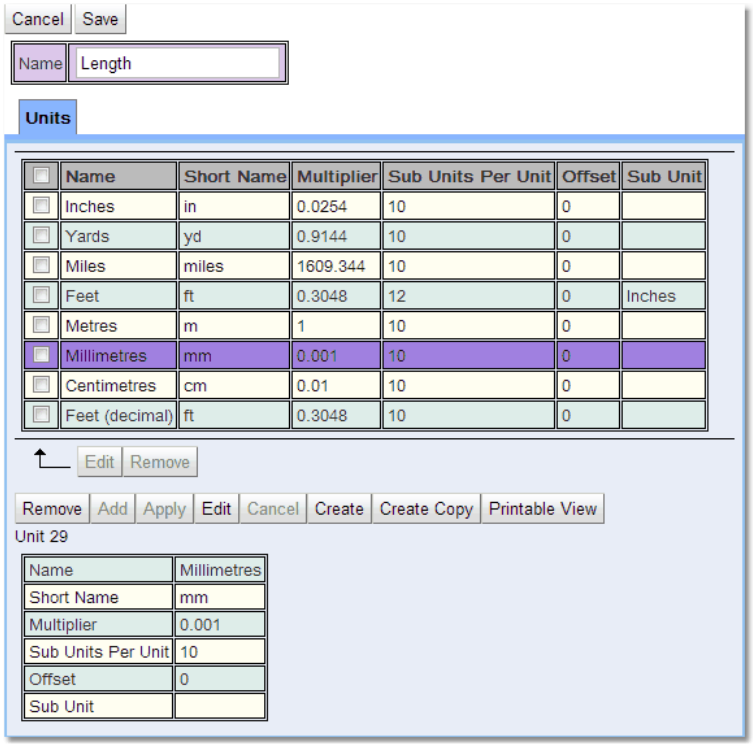
When in edit mode for **String Enum Def** to add a new value, use the **Enter New Value** box and press **Enter** when finished. The new value is added to the bottom of the list. To remove items select the check box for the items to be removed and then press the **Remove** button. To move an item to a different position in the list use the [drag] column. Select the item to be dragged and then drag and drop it where required.



© 2010-2015 Exprodo Software

Editing Units

When in edit mode there is a list view, once a unit is selected there will be a single record view underneath the list view. For more information on How to Edit Multiple Items At Once[128] in the list view and How to Edit A Single Item[130] in the record view, read the appropriate sections of the Data Explorer[127] chapter



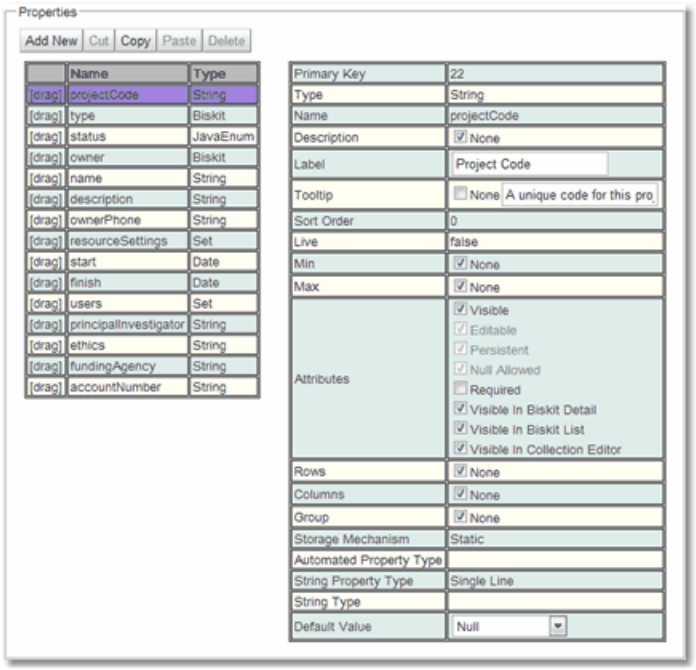
© 2010-2015 Exprodo Software

6.17.5.1 Bakery Editor Examples

6.17.5.1.1 Adding Properties

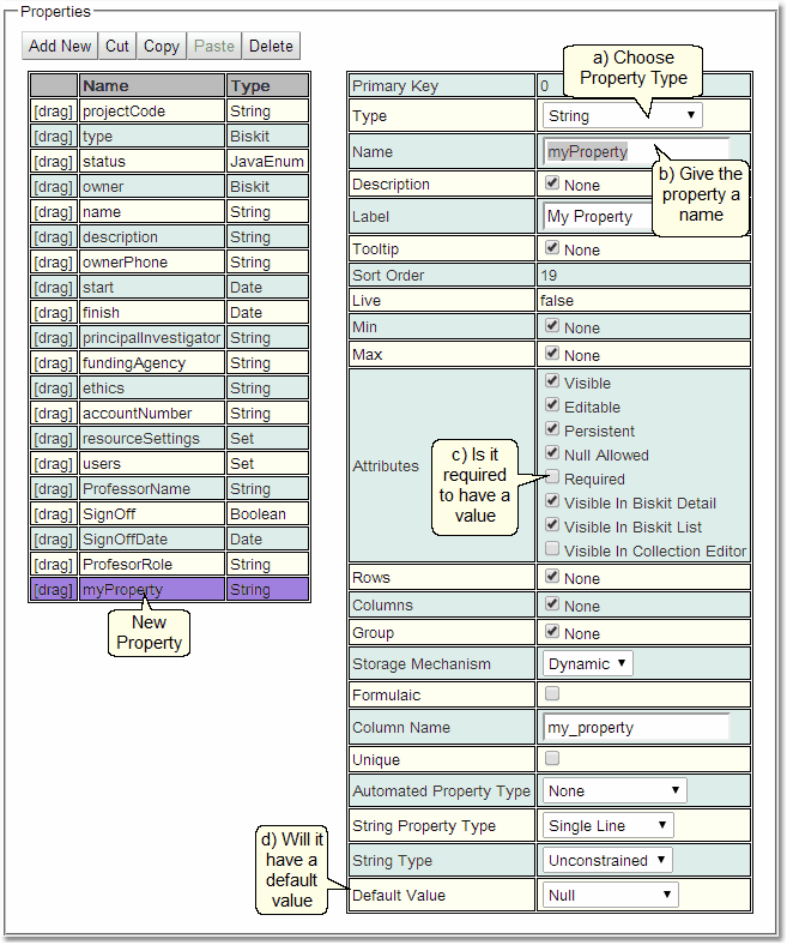
An example of adding a *property*[448] to the *Biskit*[446] of type **Project** called **ProfessorSignOff**, of type **boolean** which is required to always have a value and starts with the value **False**.

1. Go to **Admin->Bakery**,
2. Press the **Open All** button to open the **Biskit Tree**
3. Select the *Biskit* to add *properties* to (in this case **Project**)
4. Press the **Edit** button to get the *Biskit* into edit mode



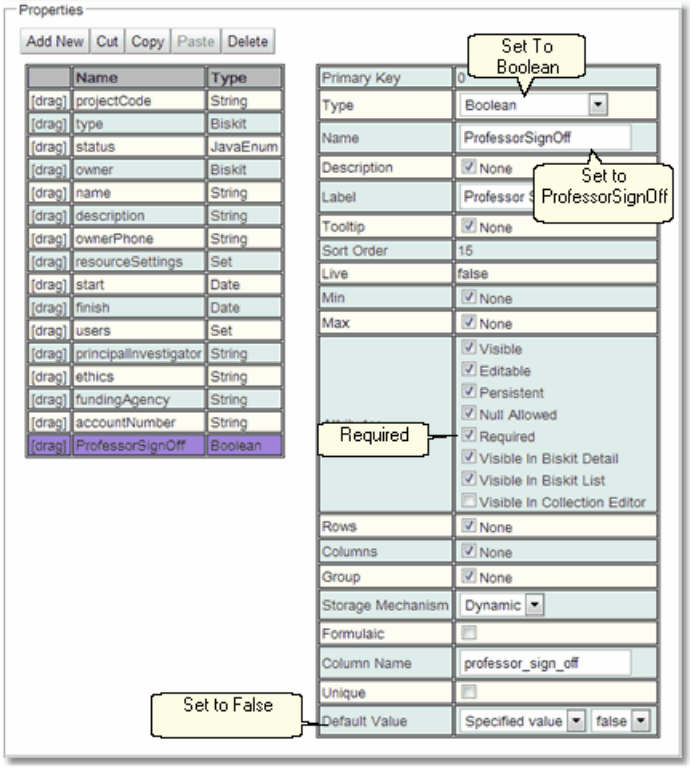
5. Press the **Add New** button to add a new *property*.

© 2010-2015 Exprodo Software

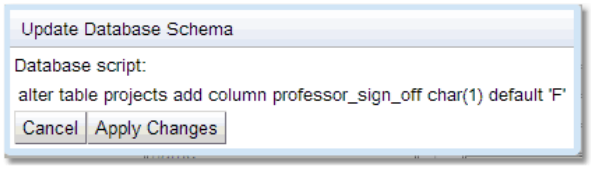


6. Update the *meta properties*[448] of the *property*. Most of them will not need to be changed but as a minimum
   a) Decide on the **Type** required.
   b) Give the *property* a name, this needs to be unique, **Calpendo** will update the **Label** and **Column Name** *meta properties* at the same time. Usually these can be left alone, but **Label** may need to be modified as this is what is displayed for this *property* in reports etc.
   c) When a *Biskit* of type **Project** is created by a user, is the user required to give this *property* a value. (**Null Allowed**, **Min** and **Max** also have an impact on data requirements and may be used instead of or in conjunction with **Required**.)
   d) Will the *property* have a default value when a *Biskit* of type **Project** is created.

© 2010-2015 Exprodo Software



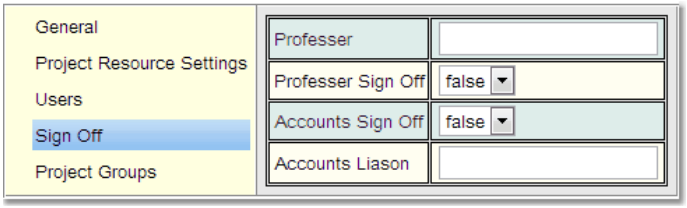
7. Press the **Save** button.
8. If there are no errors then update the database:
   a) Press **Update DB Schema** to implement changes in the DB, the following should pop up:



9. Press the **Apply Changes** button to update the database. There will be a response saying how many rows affected. Press **OK** to finish.

   a) Press **Validate Biskits** button to check the database *Biskits*.
   b) Press **Reload Database Configuration**, to load the new database into **Calpendo**
   c) Refresh the browser

Remember if adding a large number of *properties* to a *Biskit* and they need to be displayed in particular groups under specific tab names then use the **Group** *meta-property*. All *properties* with the same **Group** *meta-property* will be displayed under the same tab which will be labelled with the value of the **Group** *meta-property*, unless a layout has been defined for this *Biskit* in the **Layout Editor**. (The **Project** *Biskit* will always use **Group** for its layout).



© 2010-2015 Exprodo Software

6.17.5.1.2 Adding Formulaic Properties

An example of adding a formulaic *property*[448] to the *Biskit*[446] of type **Booking**, which will be called **BookingCost** and will calculate the cost of the *booking*[446] depending on the duration of the *booking* and whether it was cancelled or not.

If a *booking* was **Denied,** or **Cancelled** more than 7 days before it was due, then no cost. If it was **Cancelled** more than 2 but less than 7 days ago, then 20% of cost, if it was **Cancelled** with less than 2 days to go then 50% of the cost otherwise full cost.

The formula used will be:

```
case
  when [status] = 'DENIED' then 0
  when [status] = 'APPROVED' or [status] = 'REQUESTED' then
[projectResourceSettings.costPerHour] * [durationInMinutes]/60
  else
    case
      when datediff([dateRange.start], [cancelled]) >= 7 then 0
      when datediff([dateRange.start], [cancelled]) >= 2 then
0.2*[projectResourceSettings.costPerHour] * [durationInMinutes]/60
      else 0.5*[projectResourceSettings.costPerHour] * [durationInMinutes]/60
    end
end
```

[**status**] will refer to a *property* called **status** found on the **Booking** *Biskit*.

[**projectResourceSettings.costPerHour**] will refer to a *property* called **costPerHour** that is found on the *Biskit* that is *referenced*[449] by the **projectResourceSettings** *property* of the **Booking** *Biskit*. In this case the type of *Biskit referenced* is **Project Resource Settings.**

[**durationInMinutes**] will refer to a *property* called **durationInMinutes** found on the **Booking** *Biskit*.

[**dateRange.start**] will refer to a *property* called **dateRange** found on the **Booking** *Biskit*. **dateRange** is of type **DateRange**, this type has some sub types, one of which is **start**. **start** holds the date and time of the start of the date range.

[**cancelled**] will refer to a *property* called **cancelled** found on the **Booking** *Biskit*.

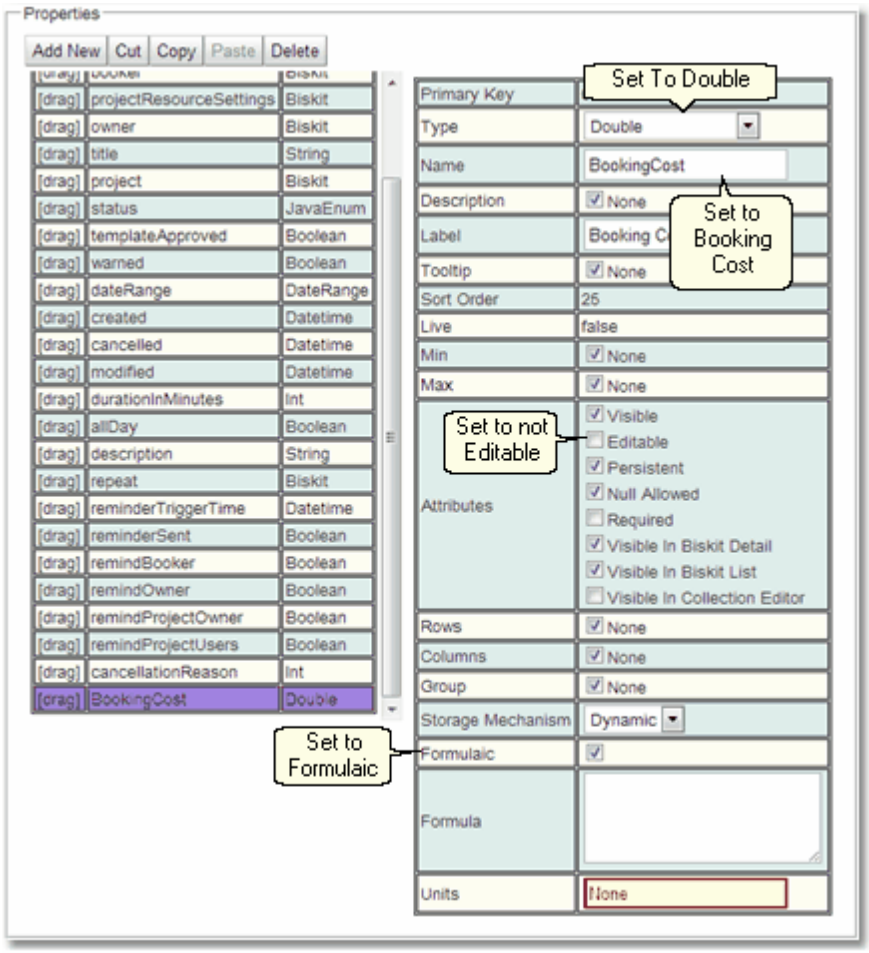
datediff is a MySQL function that returns the difference between two dates as a whole number of days. For more accurate time differences use a combination of **timediff()** and **time_to_sec()** (i.e. **time_to_sec(timediff**(date1,date2)**)** returns the number of seconds between the two dates).

We will be using the **costPerHour** *property* that can be set up for each *resource*[449] a *project*[448] can use. This way individual *projects* can be set up to pay different amounts for each *resource*.

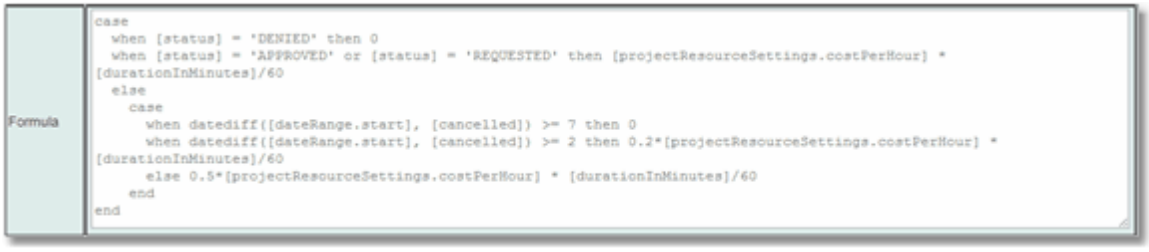
1. Go to **Admin->Bakery**,
2. Press the **Open All** button to open the **Biskit Tree**
3. Select the *Biskit* to add *properties* to (in this case **Booking**)
4. Press the **Edit** button to get the *Bislkit* into edit mode

© 2010-2015 Exprodo Software

5. Press the **Add New** button to add a new *property*.
6. Update the *meta properties* of the *property*. Most of them do not need to be changed but as a minimum
   a) Change **Type** to double.
   b) Change **Name** to **BookingCost**.
   c) Make it not **Editable**.
   d) Make it **Formulaic**.

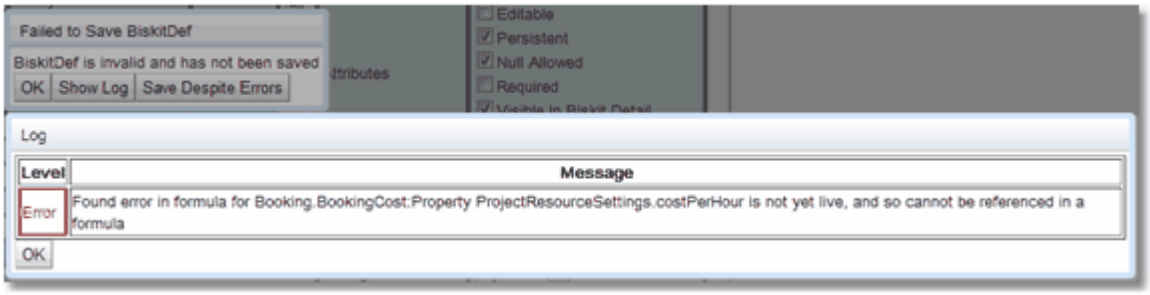


7. Enter the formula required.


Formula
case
when [status] = 'DENIED' then 0
when [status] = 'APPROVED' or [status] = 'REQUESTED' then [projectResourceSettings.costPerHour] * [durationInMinutes]/60
else
case
when datediff([dateRange.start], [cancelled]) >= 7 then 0
when datediff([dateRange.start], [cancelled]) >= 2 then 0.2*[projectResourceSettings.costPerHour] * [durationInMinutes]/60
else 0.5*[projectResourceSettings.costPerHour] * [durationInMinutes]/60
end
end

© 2010-2015 Exprodo Software

8. Press the **Save** button. An error box will appear:



Check the error by pressing **Show Log**. If the error is complaining about a *property* not being live yet then press the **OK** button, and then the **Save Despite Errors** button.

9. As this is adding a formulaic *property* we only need to reload the database configuration.
   a) Press **Reload Database Configuration**, to load the new database into **Calpendo**
   b) Refresh the browser

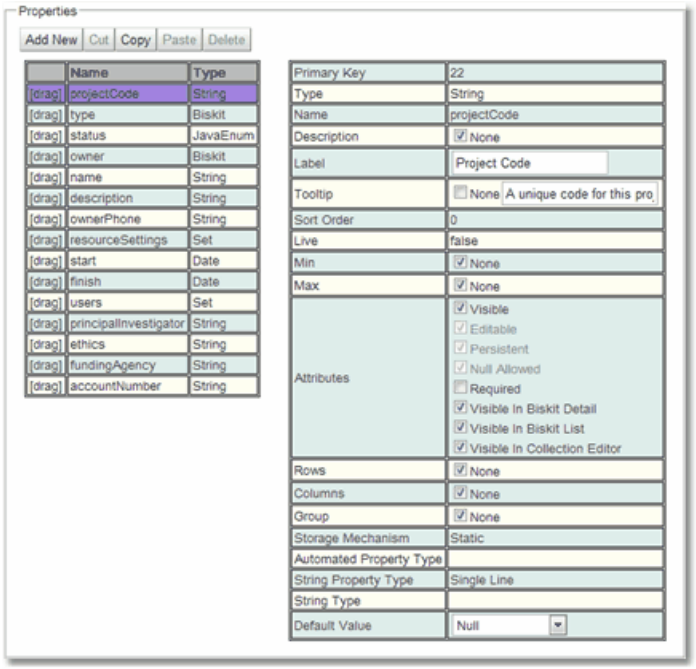
Go and check *bookings* to see if the **Booking Cost** is present. In this case if when viewing a *booking* in the *Booking Calendar*[446] at the bottom of the **Booking** will appear the **Booking Cost**.

© 2010-2015 Exprodo Software

6.17.5.1.3 Adding Properties For File Attachments

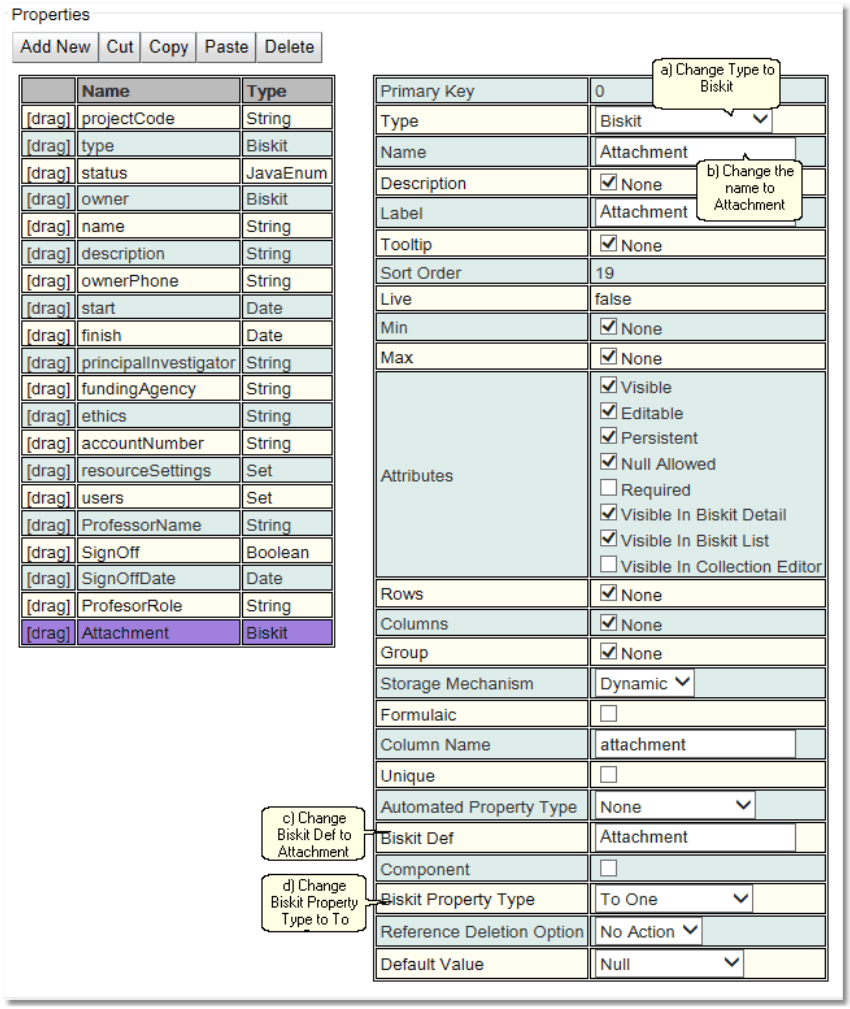
An example of adding a *property*[448] to the *Biskit*[446] of type **Project** to hold an file attachment.

1. Go to **Admin->Bakery**,
2. Press the **Open All** button to open the **Biskit Tree**
3. Select the *Biskit* to add *properties* to (in this case **Project**)
4. Press the **Edit** button to get the *Biskit* into edit mode



5. Press the **Add New** button to add a new *property*.

© 2010-2015 Exprodo Software



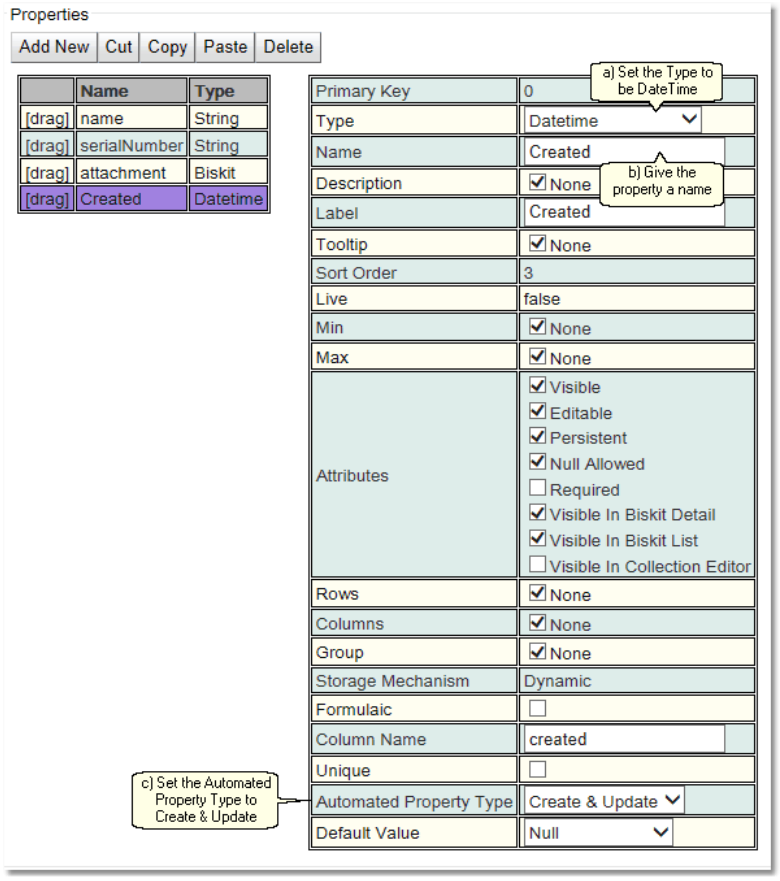
6. Update the *meta properties* 448 of the *property*.
   a) Change **Type** to **Biskit**.
   b) Give the *property* a name, in this case **Attachment.**
   c) Change **Biskit Def** to **Attachment**
   d) Change **Biskit Property Type** to **To One**.
7. Press the **Save** button.
8. If there are no errors then update the database:
   a) Press **Update DB Schema** to implement changes in the DB, a pop up will appear defining the database changes to be implemented.
9. Press the **Apply Changes** button to update the database. There will be a response saying how many rows affected. Press **OK** to finish.
   a) Press **Validate Biskits** button to check the database *Biskits*.
   b) Press **Reload Database Configuration**, to load the new database into **Calpendo**
   c) Refresh the browser

© 2010-2015 Exprodo Software

6.17.5.1.4 Adding Properties For Created/Updated/Vesion

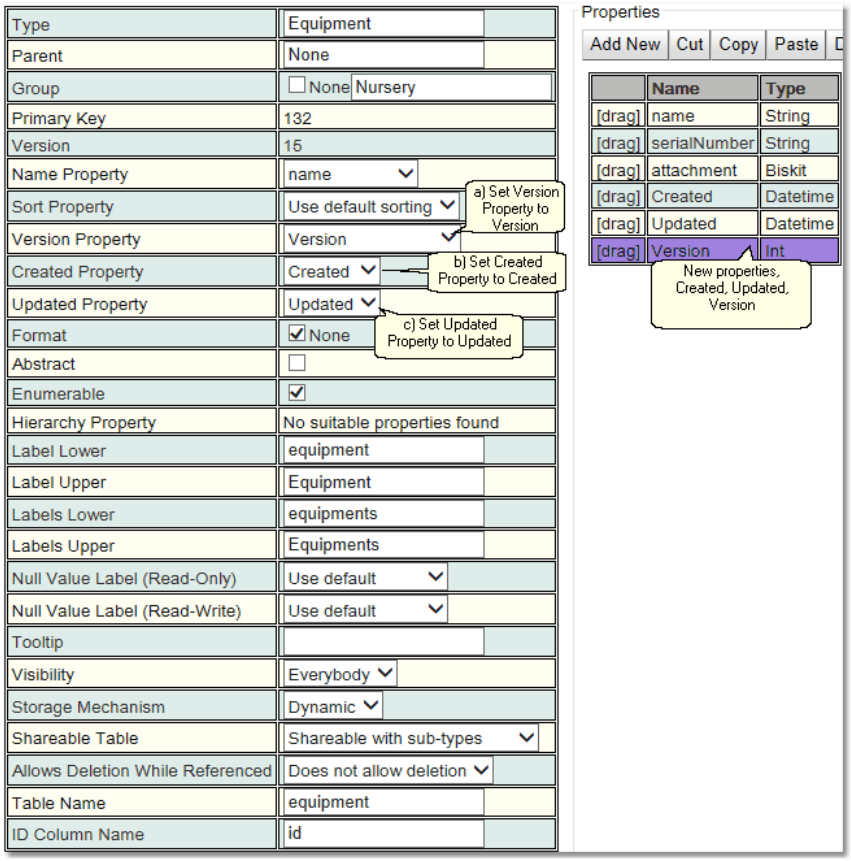
An example of adding a *properties*[448] to a *Biskit*[446] created by a user to record when a *Biskit* was created, updated and store which version is currently in use.

1. Go to **Admin->Bakery**,
2. Press the **Open All** button to open the **Biskit Tree**
3. Select the user created *Biskit* to add *properties* to
4. Press the **Edit** button to get the *Biskit* into edit mode
5. Press the **Add New** button to add a new *property*. First create a property called for example **Created**.



6. Update the *meta properties*[448] of the *property*.
   a) Change **Type** to **DateTime**.
   b) Give the *property* a name, in this case **Created.**
   c) Change **Automated Property Type** to **Create & Update**.
7. Do the same again to create a *property* for example called **Updated**.
8. And again to create a *property* for example called **Version**, this time make it of **Type Int**.

© 2010-2015 Exprodo Software



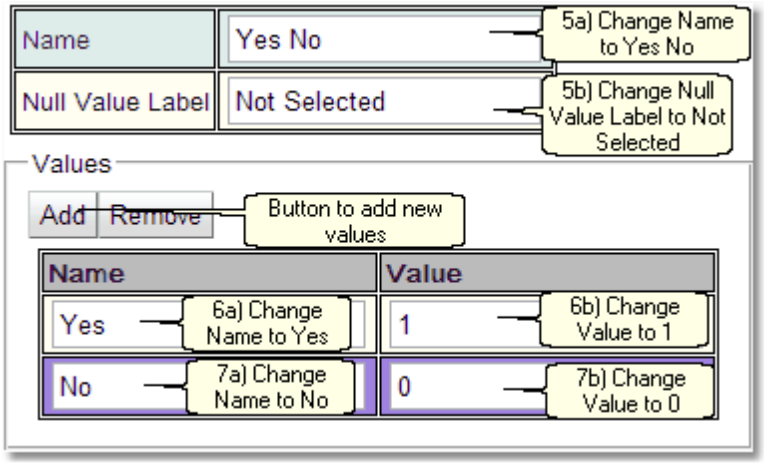
9. Set up the meta-property information
   a) Set **Version Property** meta-property to **Version**.
   b) Set **Created Property** meta-property to **Created**
   c) Set **Updated Property** meta-property to **Updated**
10. Press the **Save** button.
11. If there are no errors then update the database:
   a) Press **Update DB Schema** to implement changes in the DB,a pop up will appear defining the database changes to be implemented.
12. Press the **Apply Changes** button to update the database. There will be a response saying how many rows affected. Press **OK** to finish.
   a) Press **Validate Biskits** button to check the database *Biskits*.
   b) Press **Reload Database Configuration**, to load the new database into **Calpendo**
   c) Refresh the browser

© 2010-2015 Exprodo Software

6.17.5.1.5 Adding A New Yes-No Mapped Int Property

An example of adding a new *Mapped Int*[447] to the Database for use in integer *properties*[448], giving **Yes-No** as the drop down values rather than **True-False**.

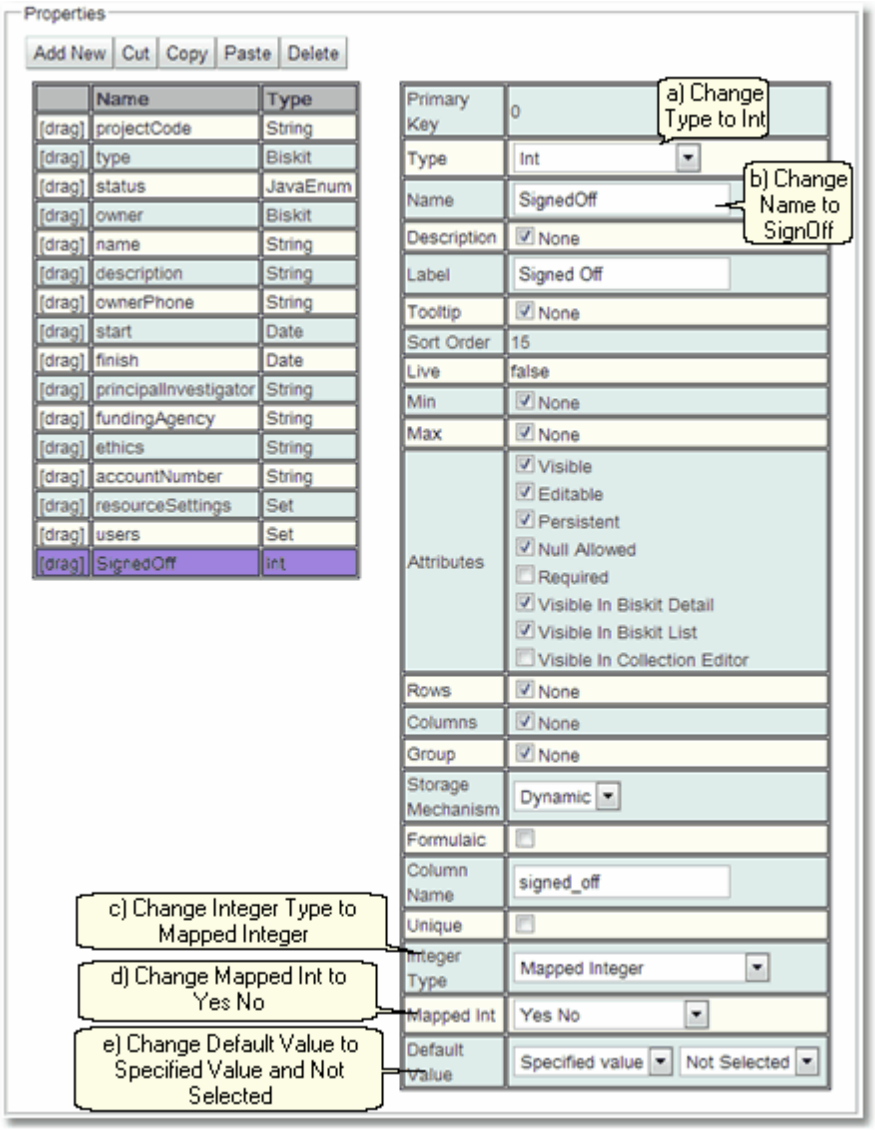
1. Go to **Admin->Bakery**,
2. Open up the **Mapped Int** tree
3. Select one of the *Mapped Int's*
4. Press the **Create** button to create a new *Mapped Int*
5. Update the *meta properties*[448].
   a) Change **Name** to **Yes No**
   b) Change **Null Value Label** to **Not Selected**
6. Use the **Add** button under **Values** to create a new value
   a) Change **Name** to **Yes**
   b) Change **Value** to **1**
7. Use the **Add** button under **Values** to create a new value
   a) Change **Name** to **No**
   b) Change **Value** to **0**



8. Press the **Save** button
9. Refresh the browser
10. Press the **Open All** button to open the **Biskit Tree**
11. Select the *Biskit*[446] to add *properties*[448] to (in this case **Project**)
12. Press the **Edit** button to get the *Biskit* into edit mode
13. Press the **Add New** button to add a new *property*.

© 2010-2015 Exprodo Software

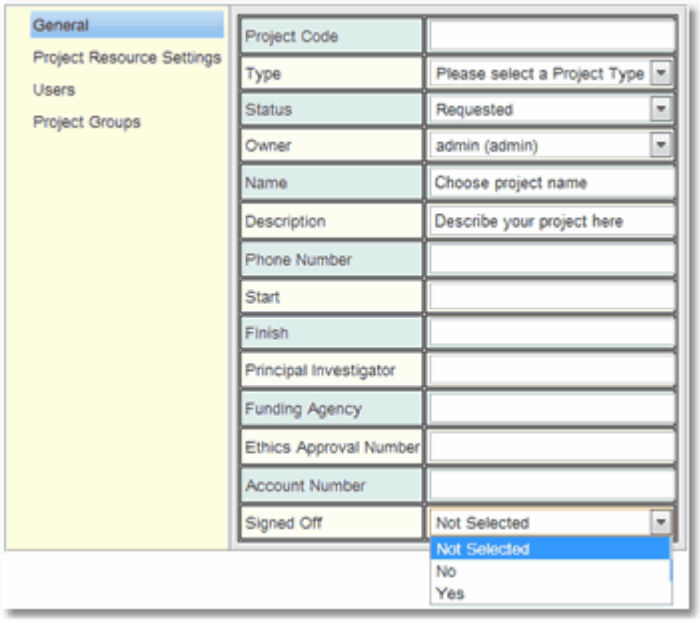
14. Update the *meta properties* of the *property*
    a) Change **Type** to **Int**.
    b) Change **Name** to something relevant (in this case **SignedOff**)
    c) Change **Integer Type** to **Mapped Integer**
    d) Change **Mapped Int** to **Yes No**.
    e) Change **Default Value** to **Specified Value** and **Not Selected**



15. Press the **Save** button. Press **Save Despite Errors** if required.
16. If there are no errors the database needs to be updated:
    a) Press **Update DB Schema** to implement changes in the DB, and then run the script to apply the changes
    b) Press **Validate Biskits** button to check the database *Biskits*
    c) Press **Reload Database Configuration**, to load the new database into **Calpendo**
    d) Refresh the browser

© 2010-2015 Exprodo Software

Looking at a **Project** *BiskitDef*[446] in the **Create Project** view we see:



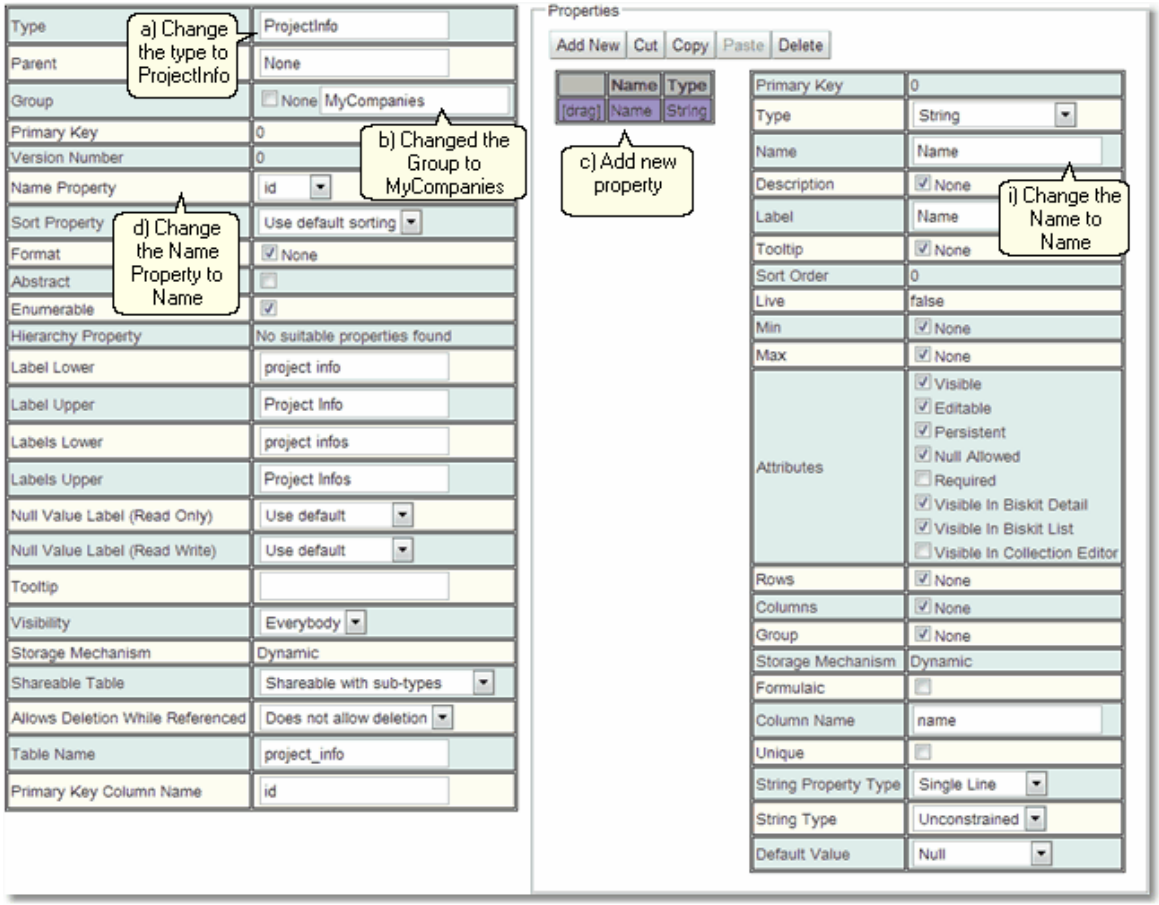
The **Default Value** could have been set up as either **No** or **Yes**, rather than **Not Selected**, also the order the values appear depends on the order of the *Mapped Int*, so if **No** has a higher value than **Yes** it will appear below **Yes.**

© 2010-2015 Exprodo Software

6.17.5.1.6 Creating A New Basic BiskitDef

An example of creating a *BisketDef*[446] called **ProjectInfo** which has one *property*[448] called **Name** which is of type **String**.

1. Go to **Admin->Bakery**.
2. Press the **Open All** button to open the **Biskit Tree**.
3. Select any *Biskit*[446].
4. Press the **Create** button to create a new *Biskit* and enter edit mode.
5. Update the *meta-properties*[448] of the *Biskit*.
   a) Change **Type** to be the new *BiskitDef* name, in this case **ProjectInfo**
   b) Change **Group** to help with finding all the new *BisketDefs* using the **Filter Biskit Group...** button
   c) In Properties **Add New**
      i. Change **Name** to **Name**
   d)Change **Name Property** on the *BisketDef* to point to the new **Name** *property*



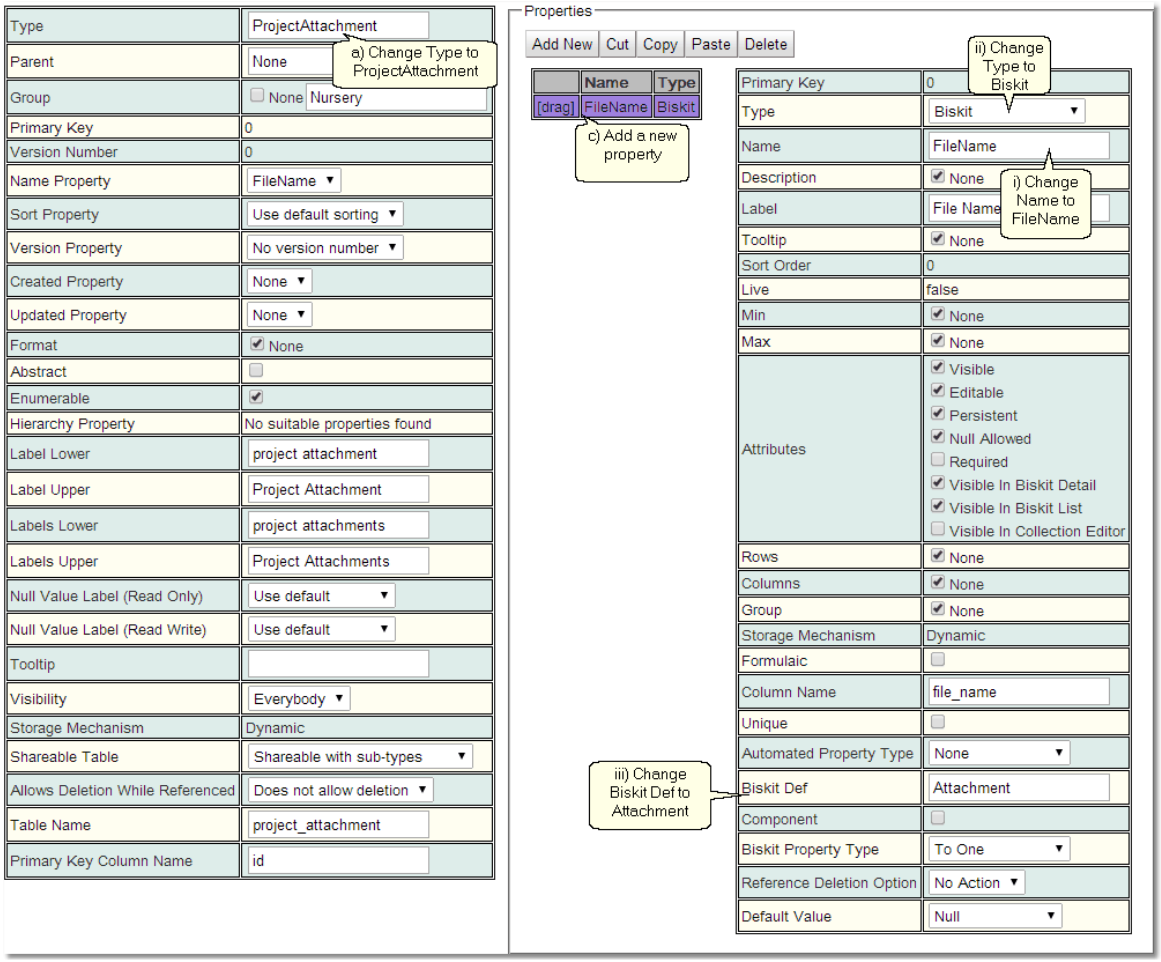
6. Press the **Save** button.
7. If there are no errors the database will need to be updated:
   a) Press **Update DB Schema** to implement changes in the DB, and then run the script to apply the changes
   b) Press **Validate Biskits** button to check the database *Biskits*.
   c) Press **Reload Database Configuration**, to load the new database into **Calpendo**
   d) Refresh the browser

© 2010-2015 Exprodo Software

6.17.5.1.7 Creating A Set Of BiskitDef (Attachments)

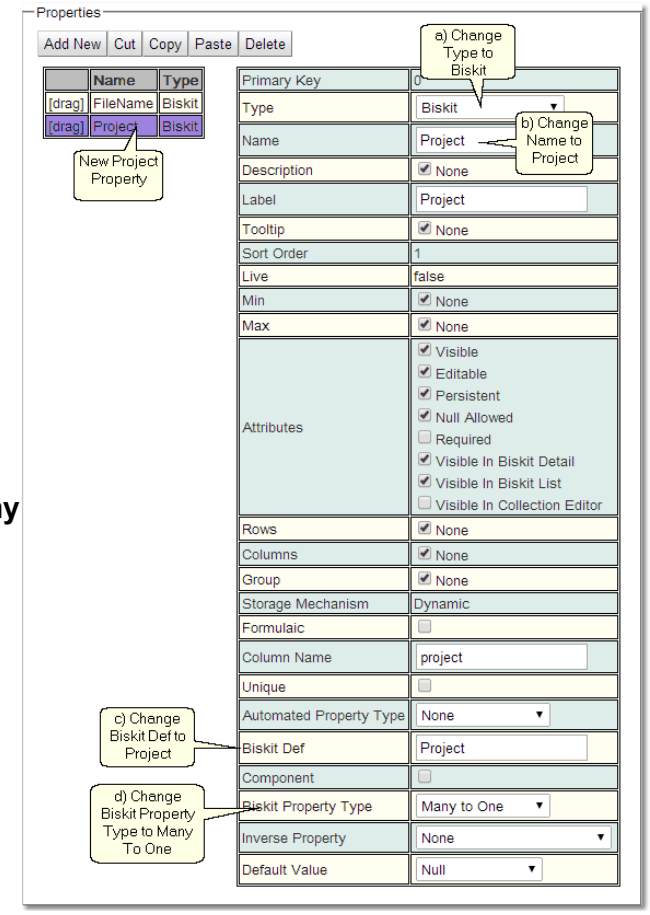
An example of creating a Set of *BiskitDef*[446], for this example we will create a Set of Attachments, to store a number of documents.

1. Go to **Admin->Bakery**.
2. Press the ***Open All*** button to open the **Biskit Tree**.
3. Select any *Biskit*[446].
4. Press the **Create** button to create a new *Biskit* and enter edit mode. First we need to create the *BiskitDef* to hold the attachments.
5. Update the *meta-properties*[448] of the *Biskit*.
   a) Change **Type** to be the new *BiskitDef* name, in this case **ProjectAttachment**.
   b) Change **Group** to help find all the new *BisketDefs* using the **Filter Biskit Group...** button
   c) In **Properties Add New**
      i. Change **Name** to **FileName**
      ii. Change **Type** to **Biskit**
      iii. Change **Biskit Def** to **Attachment**

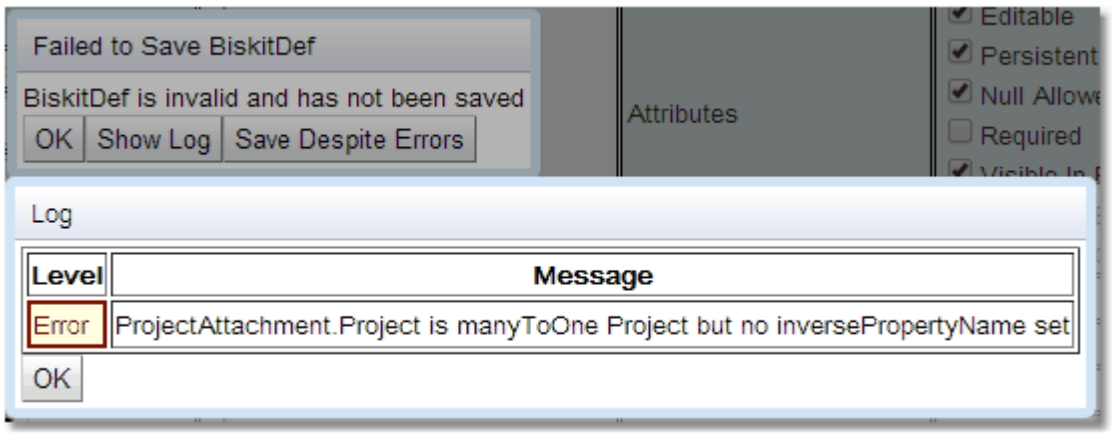


© 2010-2015 Exprodo Software

6. Create a new *property*.
   a) Change **Type** to **Biskit**.
   b) Change **Name** to **Project**
   c) Change **Biskit Def** to **Project**
   d) Change **Biskit Property Type** to **Many To One**.



7. Press the **Save** button. There will be an error:

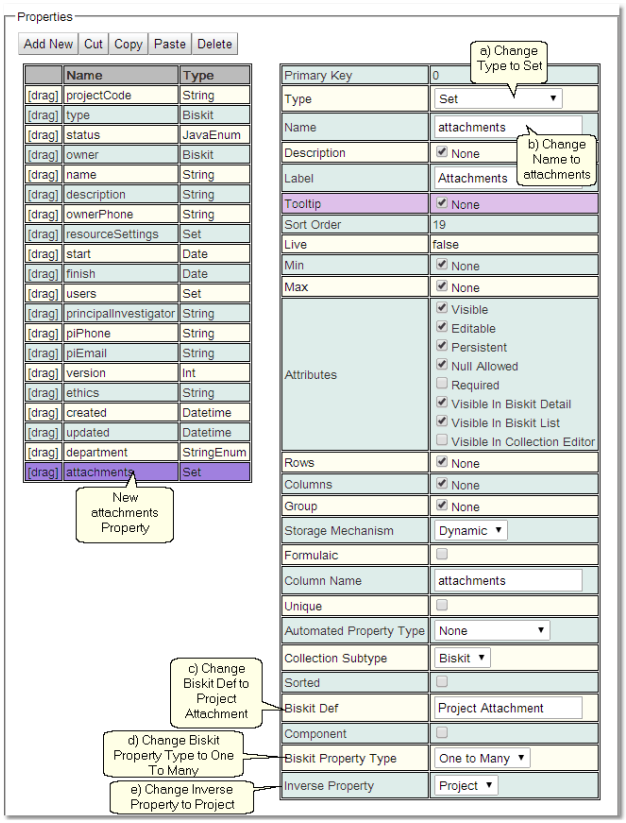


Check the error by pressing **Show Log**. If the error is complaining about a no inversePropertyName being set press the **OK** button, and then the **Save Despite Errors** button.

© 2010-2015 Exprodo Software

8. Select the **Project** *BiskitDef,* press the **Edit** button to get back into edit mode.

9. In **Properties Add New**.
   a) Change **Type** to **Set**.
   b) Change **Name** to **attachments**
   c) Change **Biskit Def** to **Project Attachment**
   d) Change **Biskit Property Type** to **One To Many**
   e) Change **Inverse Property** to **Project**

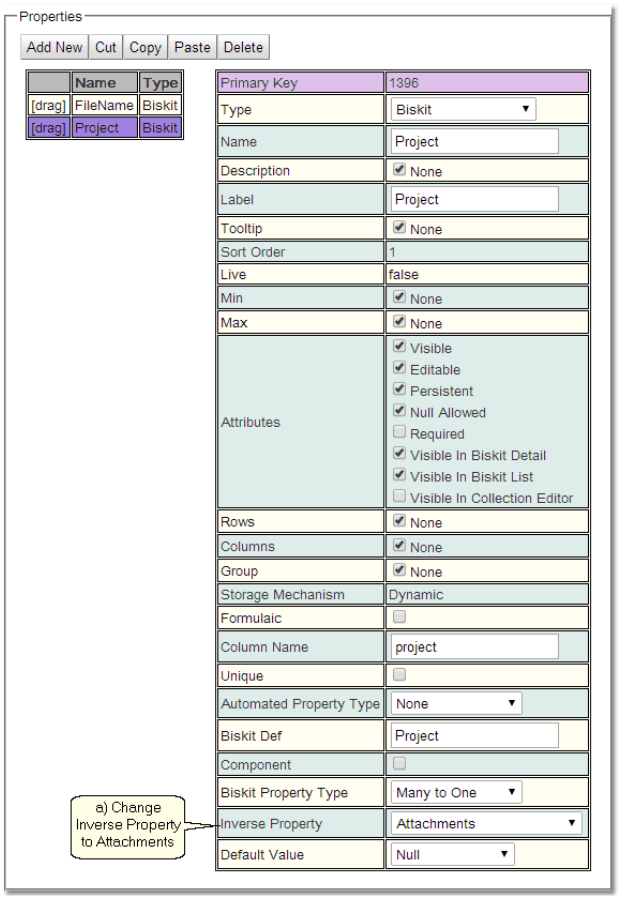


10. Press the **Save** button, and this should save without an error.

© 2010-2015 Exprodo Software

11. Select the **Project Attachment** *BiskitDef,* press the **Edit** button to get back into edit mode.

12. Select the **Project** *property*.
    a) Change **Inverse Property** to **Attachments**.



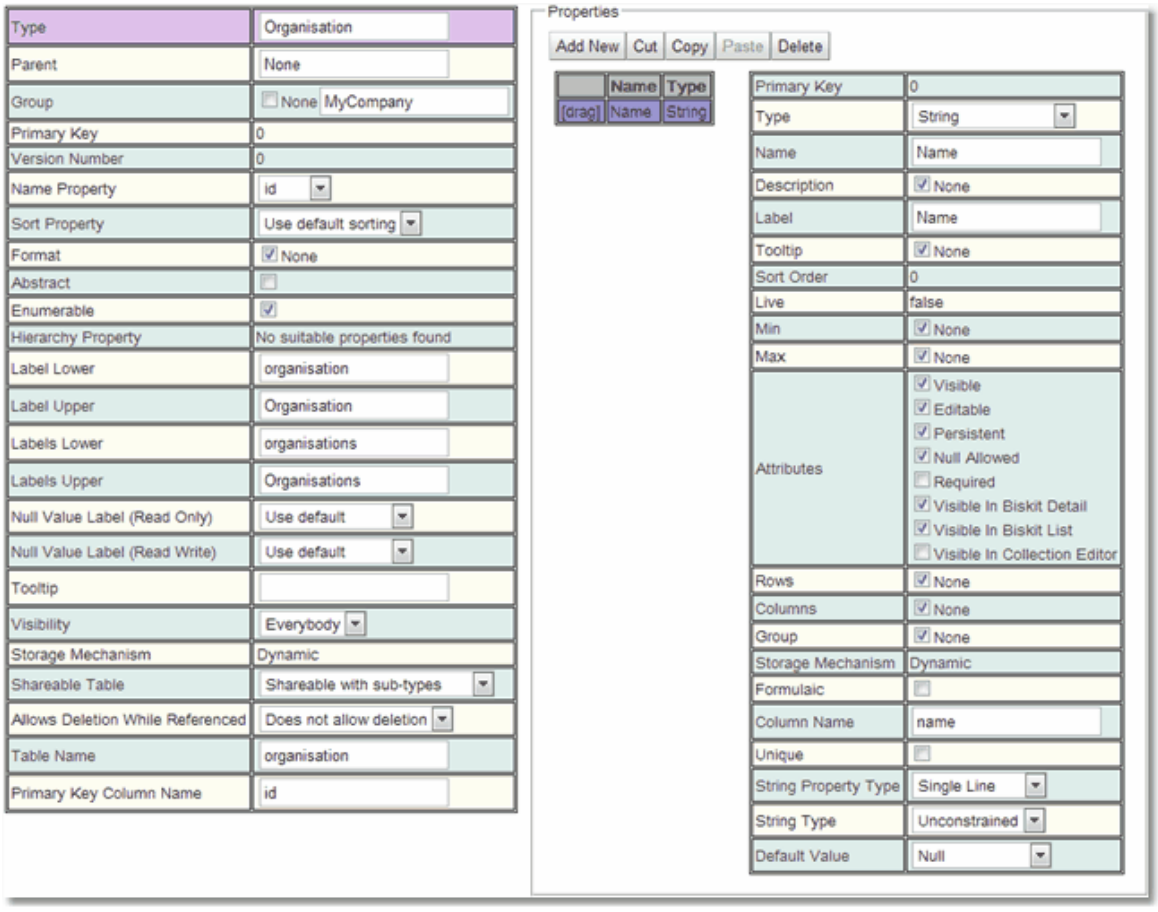
13. Press the **Save** button.
14. If there are no errors the database will need to be updated:
    a) Press **Update DB Schema** to implement changes in the DB, and then run the script to apply the changes
    b) Press **Validate Biskits** button to check the database *Biskits*.
    c) Press **Reload Database Configuration**, to load the new database into **Calpendo**
    d) Refresh the browser

© 2010-2015 Exprodo Software

6.17.5.1.8 Creating A Hierarchy Of BiskitDef

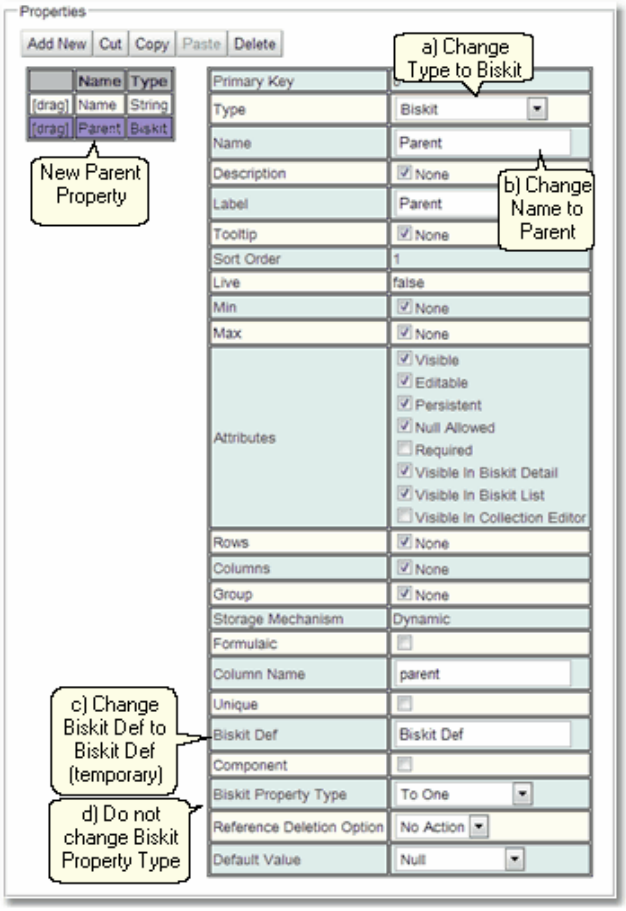
An example of creating a hierarchy of *BiskitDef*[446] similar to **Location**. Where a parent can have many children of the same type of *BiskitDef* and a child will have only one parent. We will create one called **Organisation**.

1. Go to **Admin->Bakery**.
2. Press the ***Open All*** button to open the **Biskit Tree**.
3. Select any *Biskit*[446].
4. Press the **Create** button to create a new *Biskit* and enter edit mode.
5. Update the *meta-properties*[448] of the *Biskit*.
   a) Change **Type** to be the new *BiskitDef* name, in this case **Organisation**.
   b) Change **Group** to help find all the new *BisketDefs* using the **Filter Biskit Group...** button
   c) In **Properties Add New**
      i. Change **Name** to **Name**
   d)Change **Name Property** on the *BisketDef* to point to the new **Name** *property*[448]



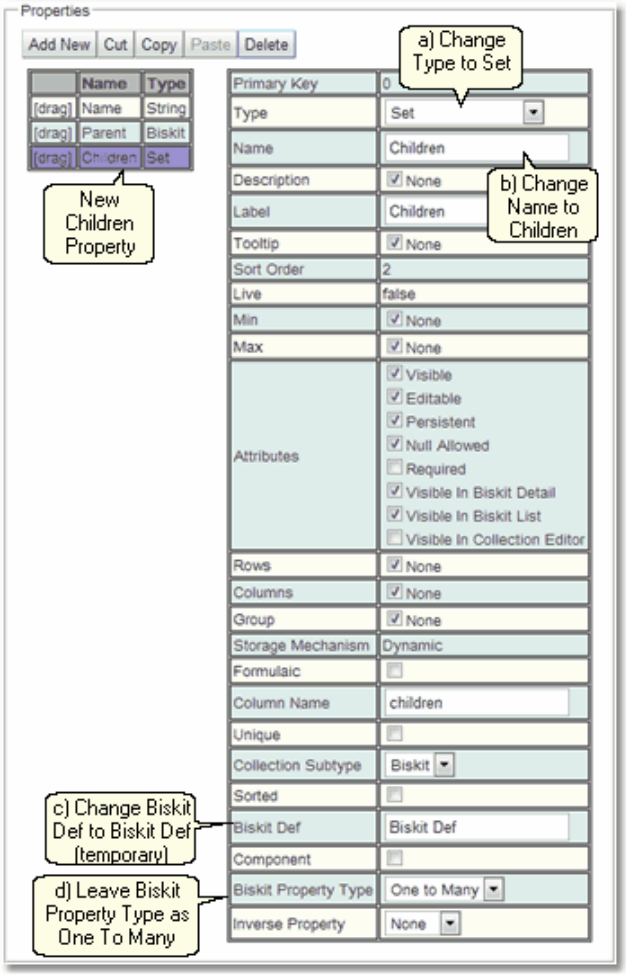
© 2010-2015 Exprodo Software

6. Create a new *property*.
   a) Change **Type** to **Biskit**.
   b) Change **Name** to **Parent**
   c) Change **BiskitDef** to *BiskitDef* (**Organisation** is not know yet because we have not saved so use this as a holder)
   d) Leave **Biskit Property Type** as **To One**.

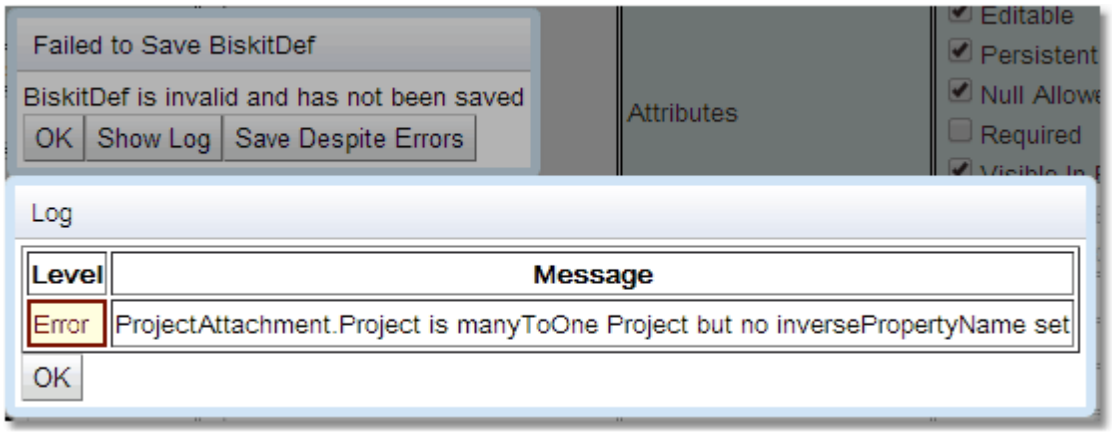


© 2010-2015 Exprodo Software

7. Create a new *property*
   a) Change **Type** to **Set**.
   b) Change **Name** to **Children**
   c) Change **BiskitDef** to *BiskitDef* (**Organisation** is not know yet because we have not saved so use this as a holder)
   d) Leave **Biskit Property Type** as **One To Many**



8. Press the **Save** button. There will be an error:

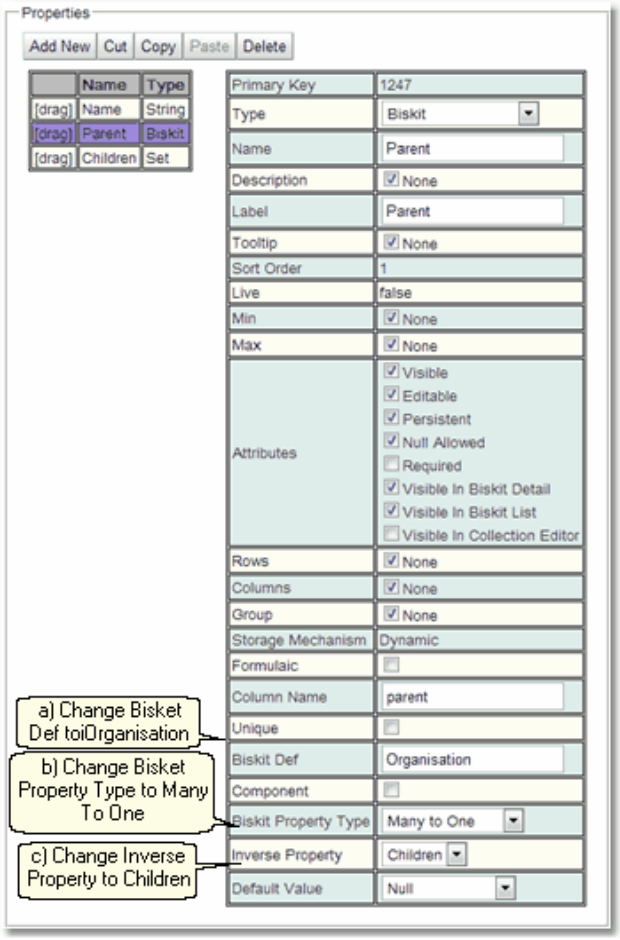


Check the error by pressing **Show Log**. If the error is complaining about a *property* not having the correct *BisketDef* press the **OK** button, and then the **Save Despite Errors** button.

9. Press the **Edit** button to get back into edit mode.

© 2010-2015 Exprodo Software

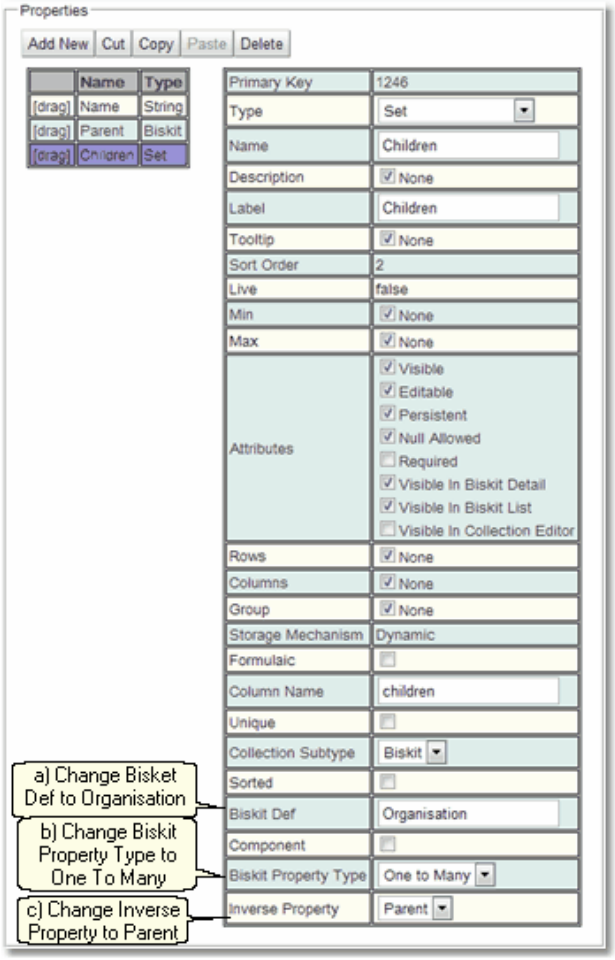
*10.*Select the **Parent** *property*
a) Change **BisketDef** to **Organisation**.
b) Change **Biskit Property Type** to **Many To One**
c) Change **Inverse Property** to **Children**



11.Press the **Save** button, there will be an error, **Save Despite Error**. This is needed so that when the **Children** property is set up it finds the completed **Parent** *property* to set up **Inverse Property**
12.Press the **Edit** button to get back into edit mode.

© 2010-2015 Exprodo Software

13. Select the **Children** *property*.
    a) Change **BisketDef** to **Organisation**.
    b) Change **Biskit Property Type** to **One To Many**
    c) Change **Inverse Property** to **Parent**



14. Change the **Hierarchy Property** *meta-property* of the **Organisation** *Biskit* to be **children**.
15. Press the **Save** button.
16. If there are no errors the database will need to be updated:
    a) Press **Update DB Schema** to implement changes in the DB, and then run the script to apply the changes
    b) Press **Validate Biskits** button to check the database *Biskits*.
    c) Press **Reload Database Configuration**, to load the new database into **Calpendo**
    d) Refresh the browser

© 2010-2015 Exprodo Software

6.17.5.1.9 Creating A Master-Slave Biskit Relationship

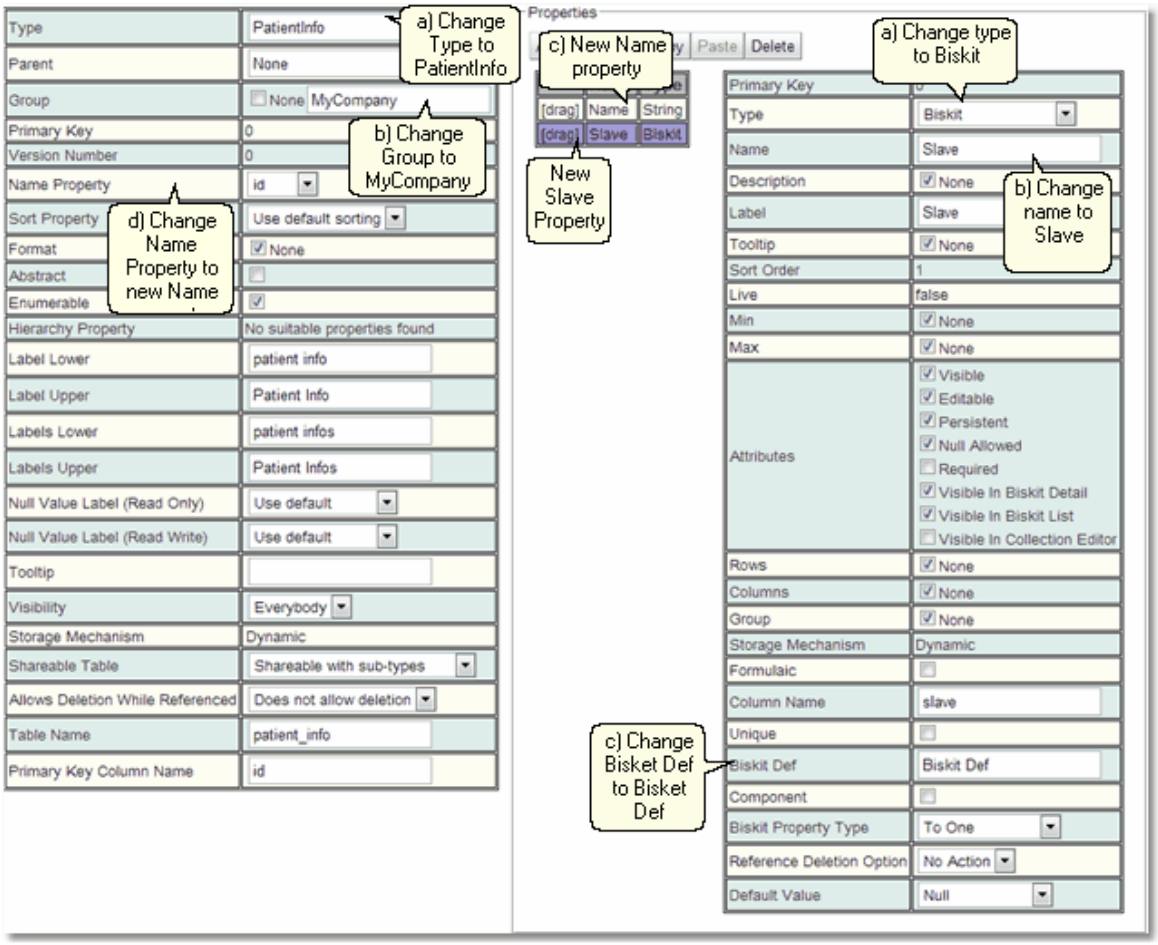
An example of creating a **Master-Slave** *Biskit*[446] relationship. This type of relationship is needed if there are going to be a large number of pieces of data to be stored in effectively one table. MySQL has a limit of about 1000 columns to a table, so if more are needed, *Biskits* that are completely linked together will need to be created. Where if a record is created in one an equivalent record will be created in the other and they will be referenced through the same key.

The **Master-Slave** that will be created will have only one *property*[448] on the **Master** (**Name** of type **String**), once this has been created then all the *properties* needed, can be added to each of the *Biskits*. A **Master** could have many **Slave** *Biskits* as required, a *property* needs to be set up on the **Master** to point to each **Slave**. Each **Slave** just needs one *property* to point to the **Master**. The **Primary Key Column Name** in the **Slave** MUST be different to that in the **Master**. If there are multiple **Slaves** then the **Primary Key Column Name** in each **Slave** MUST be different.

1. Go to **Admin->Bakery**.
2. Press the **Open All** button to open the **Biskit Tree**.
3. Select any *Biskit*.
4. Press the **Create** button to create a new *Biskit* and enter edit mode.

© 2010-2015 Exprodo Software

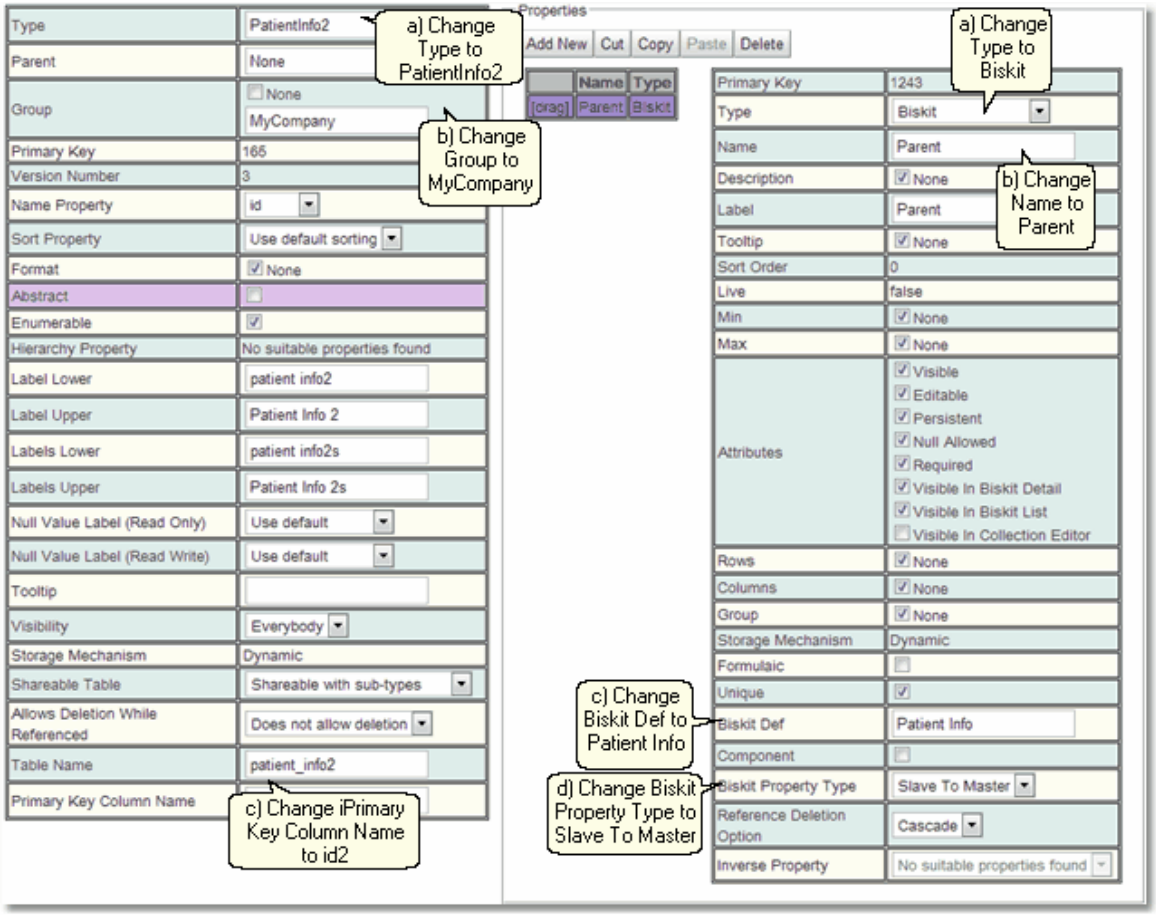
5. Update the *meta-properties*[448] of the *Biskit*.
   a) Change **Type** to be the new *BiskitDef*[446] name, in this case **PatientInfo**.
   b) Change **Group** to make it easy *to* find all the new *BisketDefs* using the **Filter Biskit Group...** button
   c) In **Properties Add New**
      i. Change **Name** to **Name**
   d)Change **Name Property** on the *BisketDef* to point to the new **Name** *property*
6. Create a new *property*.
   a) Change **Type** to **Biskit**.
   b) Change **Name** to **Slave**.
   c) Change **Biskit Def** to **Biskit Def** (The other *BisketDef* is not know yet because we have not created it so use this as a holder).



7. Press the **Save** button.

© 2010-2015 Exprodo Software

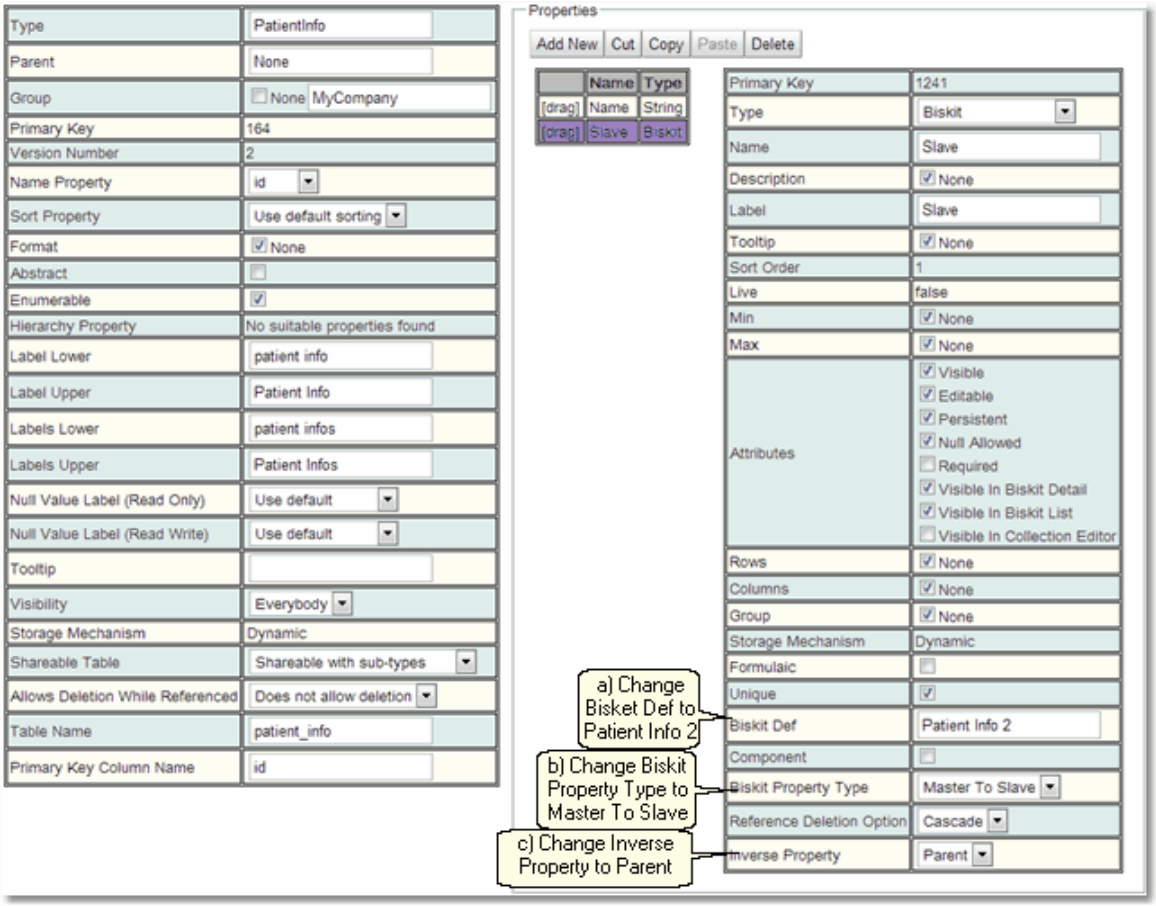
8. Press the **Create** button to create a new *Biskit* and enter edit mode.
9. Update the *meta-properties* of the *Biskit*.
   a) Change **Type** to be the new *BiskitDef* name, in this case **PatientInfo2**
   b) Change **Group** to make it easy *to* find all the new *BisketDefs* using the **Filter Biskit Group...** button
   c) Change **Primary Key Column Name** to id1 (must be different from the **Master** *BiskitDef* **Primary Key Column Name**)
10. Create a new *property*.
   a) Change **Type** to **Biskit**.
   b) Change **Name** to **Parent**
   c) Change **BiskitDef** to **Patient Info**
   d) Change **Biskit Property Type** to **Slave To Master**



11. Press the **Save** Button. There will be an error, **Save Despite Errors**.

© 2010-2015 Exprodo Software

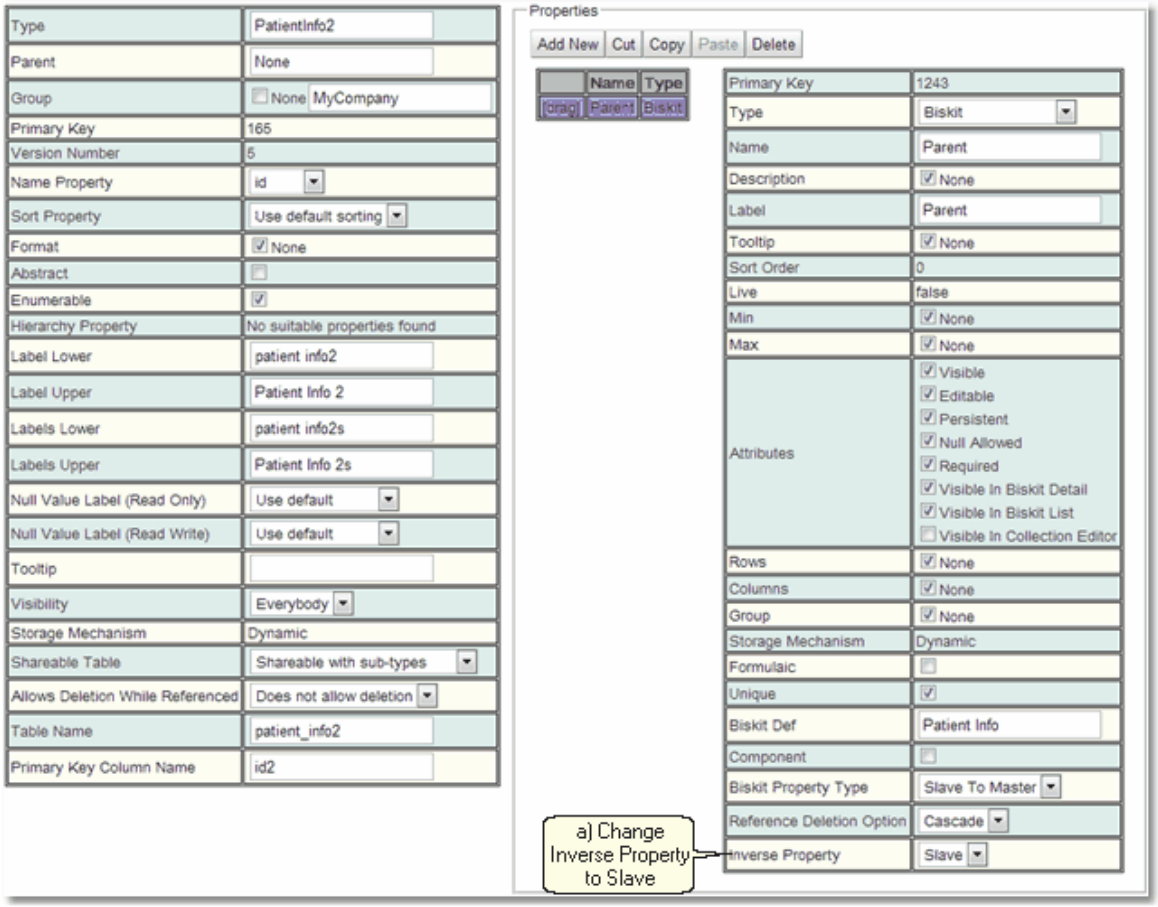
*12.*Select the **PatientInfo** *Biskit* and press the **Edit** button to get back into edit mode.
13.Select the **Slave** *property*.
a) Change **Bisket Def** to **PatientInfo2**.
b) Change **Biskit Property Type** to **Master To Slave**
c) Change **Inverse Property** to **Parent**



*14.*Press the **Save** button. There will be an error, **Save Despite Errors**.

© 2010-2015 Exprodo Software

15. Select the **PatientInfo2** *Biskit* and press the **Edit** button to get back into edit mode.
*16.*Select the **Parent** *property*.
a) Change **Inverse Property** to **Slave**



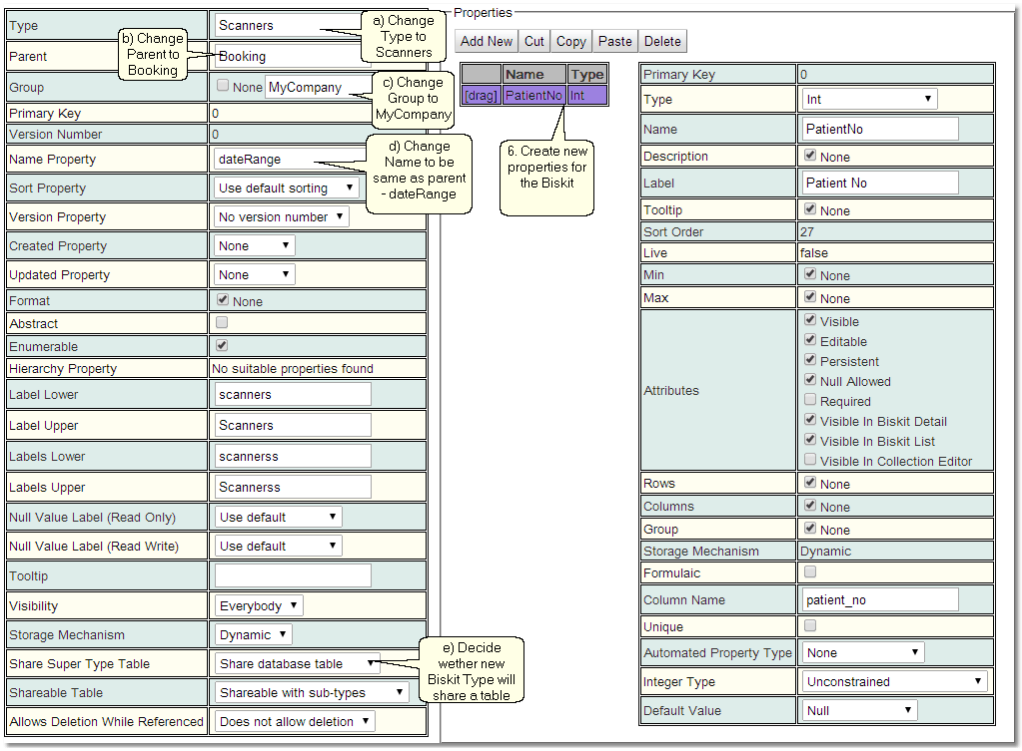
17.Press the **Save** button.
18.If there are no errors the database will need to be updated:
a) Press **Update DB Schema** to implement changes in the DB, and then run the script to apply the changes
b) Press **Validate Biskits** button to check the database *Biskits*
c) Press **Reload Database Configuration**, to load the new database into **Calpendo**
d) Refresh the browser

© 2010-2015 Exprodo Software

6.17.5.1.10 Creating An Inheriting BiskitDef (Booking)

An example of creating a *Biskit*[446] that inherits from another *Biskit*. In this case **Resource**. Please note that when editing the new *Biskit* the user cannot see the *properties*[448] that it inherits from the parent *Biskit*.

1. Go to **Admin->Bakery**.
2. Press the **Open All** button to open the **Biskit Tree**.
3. Select any *Biskit*.
4. Press the **Create** button to create a new *Biskit* and enter edit mode.
5. Update the *meta-properties*[448] of the *Biskit*.
   a) Change **Type** to be the new *BiskitDef*[446] name, in this case **ExtendedResource**.
   b) Change **Parent** to **Resource**
   c) Change **Group** to make it easy *to* find all the new *BisketDefs* using the **Filter Biskit Group...** button
   d) Decide whether new *BiskitDef* will share a table with its parent by changing **Share Super Type Table** (it is best if they do share a table).
   e) In **Properties Add New**
      i. Change **Name** to **LastBreakdown**
      ii. Change **Type** to **Date**
   f) Change **Name** *property* on the *BiskitDef* to point to the **Name** *property* of the parent



6. Press the **Save** button.
7. If there are no errors the database will need to be updated:
   a) Press **Update DB Schema** to implement changes in the DB, and then run the script to apply the changes
   b) Press **Validate Biskits** button to check the database *Biskits*
   c) Press **Reload Database Configuration**, to load the new database into **Calpendo**
   d) Refresh the browser

© 2010-2015 Exprodo Software

6.18 Layout Editor

The **Layout Editor** allows the user to define how the *properties* 448 of a *Biskit* 446 will be displayed. This is required because *Biskits* could have a large number of *properties*, numbering in the hundreds, and in order for users to find those *properties* to update them, it is necessary to organise them.

The **Layout Editor** allows the user to organise the *properties* of a *Biskit* into multiple **Property Groups**, **Property Groups** may themselves contain child **Property Groups**.

Property Groups at the same level may be organised in one of four ways, **Vertical Tabs**, **Vertical Tabs (Sorted by Name)**, **Horizontal Tabs** and **Captions**. **Property Groups** may have a heading and may also be hidden.

The *properties* contained within a **Property Group** may also be organised in one of four ways, **Simple Table**, **Multi Column Table**, **Radio Button Table** and **Custom HTML**.

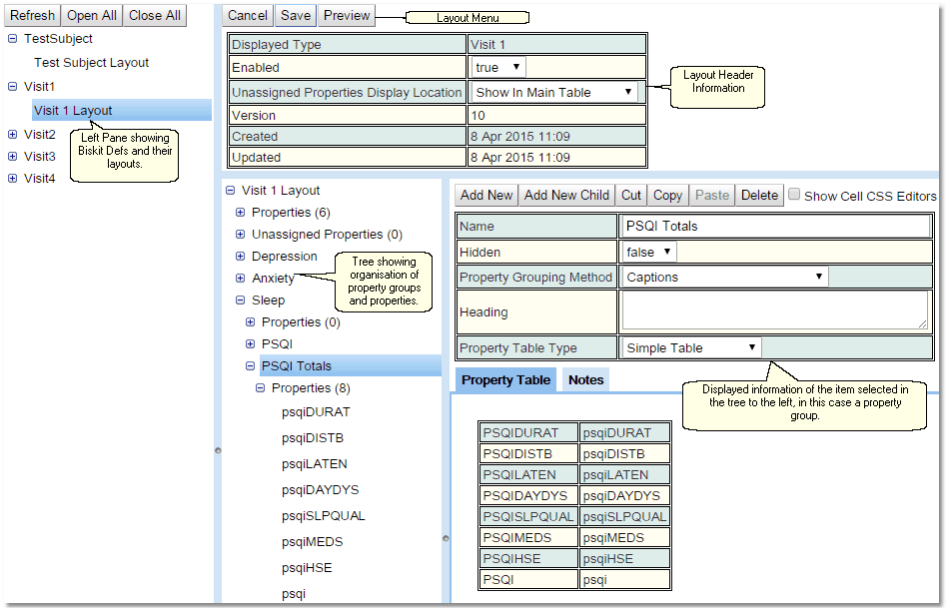
For each table, row, column or cell the standard *CSS* can be changed allowing the user to set such things as font, background colour, border size, type and colour and many other attributes.

Finally for any **Property Group, Notes** can be set up to be created and then displayed in a **Notes Waterfall**.

© 2010-2015 Exprodo Software

The **Layout Editor** is split into four sections in two panes.

1. The left hand pane shows a list of *BiskitDefs* 446 that have had layouts created for them, when a *BiskitDef* is expanded all the layouts defined for that *Biskit* will be shown. Any in red are currently not enabled.
2. The right pane is split into three areas:
   a. Top is the layout menu and the header information for the current layout. (**Top Pane**)
   b. Bottom left shows the organisation of the **Property Groups** and *properties* for this layout in the form of a tree. (**Organisation Pane**)
   c. Bottom right displays information depending on what is selected in the organisation tree in the bottom left pane. (**Information Pane**)
      i. If a **Property Group** is selected then the header information for that **Property Group** is viewed along with any *properties* assigned to that **Property Group**.
      ii. If a **Properties** branch is selected then the *properties* are shown in a list
      iii. If a single *property* is selected then information about that *property* will be shown.



© 2010-2015 Exprodo Software

Left Pane

The Left Pane shows the **Layout Tree** (see image above). Showing for each *BiskitDef* the layouts that are available. It also has a tool bar which allows the user to open and close all the branches to the tree.

For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition 96 chapter.

Top Pane With Layout Toolbar

The **Top Pane** shows the header information for a layout. To create a new layout press the **Create** button and then choose the *BiskitDef* the layout is for.

Property	Description
Displayed Type	The *BiskitDef* the layout is for.
Enabled	Whether this layout is currently to be used. Disabled layouts are shown in the layout tree in the left pane in red.
Unassigned Properties Display Location	Specifies how all unassigned *properties* will be displayed. Either ***S*how in Main Table** where the *properties* are shown in a table above the tabs section or **Shown in Property Group** where the *properties* are shown as the first tab in the tabs section with a name for the tab assigned by the user.

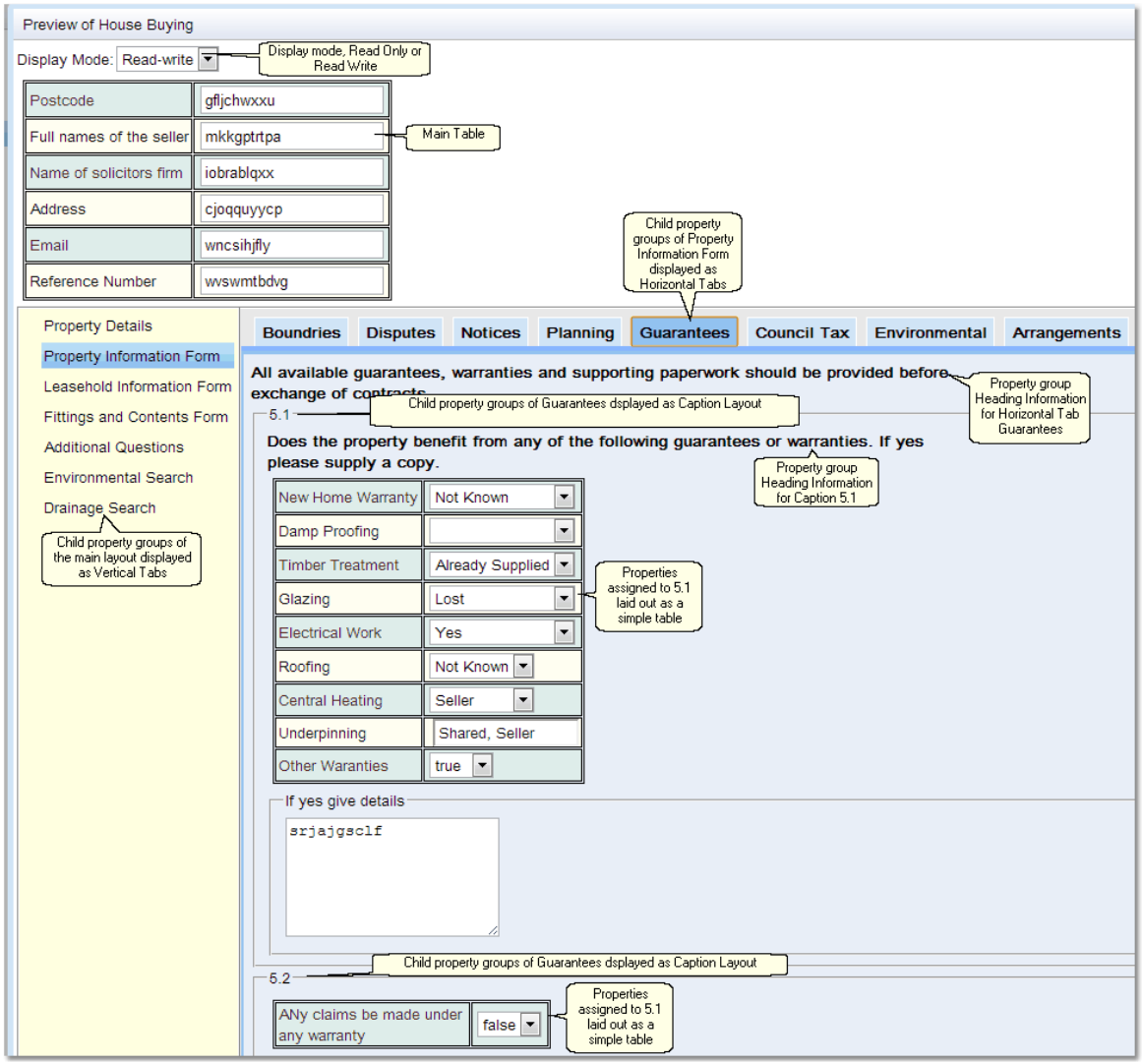
The **Top Pane** also has the **Layout** toolbar which enables the user to edit, create and delete layouts as well as being able to preview the layout already defined.

For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition 96 chapter.

Layout Toolbar Item	Description
Preview	This will display a preview of the current layout as defined. When in edit mode this preview will be up to date and you do not need to save first. The preview will display initially in read-write mode, but the user can change the mode using a toggle at the top of the view to see what the display would look like in read-only mode. Some information is only displayed in read-only mode. The information in the *properties* in the preview is constructed in a random manner to give the user an idea of what the layout would look like with data and is not read from the database.

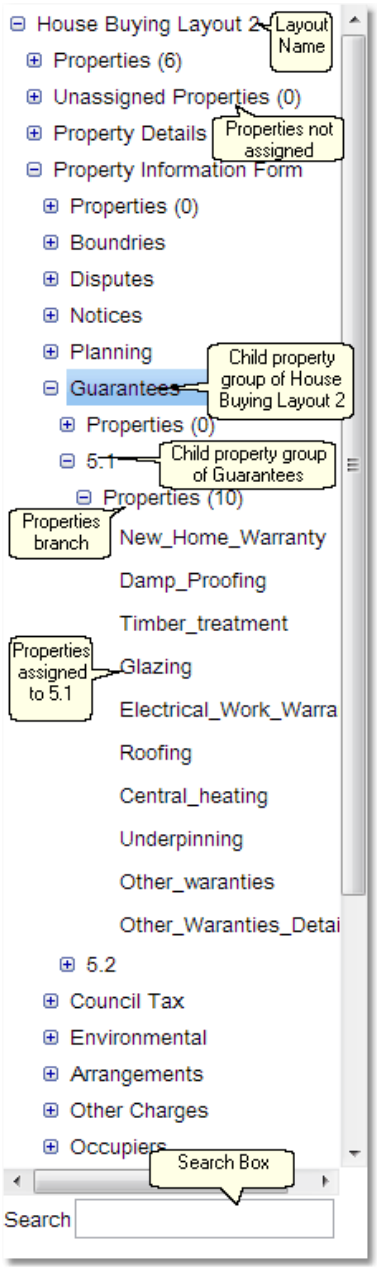
© 2010-2015 Exprodo Software

By pressing the **Preview** button a view of the current layout showing the different areas will appear. Use this preview to walk around the layout checking how it works.



© 2010-2015 Exprodo Software

Organisation Pane



This pane shows how the *properties* of a *Biskit* have been organised into **Property Groups** and child **Property Groups** as a tree structure.

The layout is defined at the top of the pane, the layout will have a **Properties** and **Unassigned Properties** branch.

Property Groups are then created When a **Property Group** is created it will automatically have a **Properties** branch which will initially be empty.

Property Groups can be reordered by dragging and dropping the group elsewhere within its parent.

To move a **Property Group** from its current parent to another parent they can be dragged and dropped or by using cut and paste. Cut the **Property Group**, select the new parent, paste the **Property Group**. Then reorder as above

The user in edit mode can move *properties* from one **Properties** branch to another by dragging and dropping. All the *properties* in a **Properties** branch can be moved by dragging the branch to a different **Properties** branch. Also *properties* can be dragged from a property list in the **Information Pane** and dropped into a **Properties** branch in the **Organisation Pane**.

Properties can be reordered within a **Property Group** by dragging and dropping the *properties*.

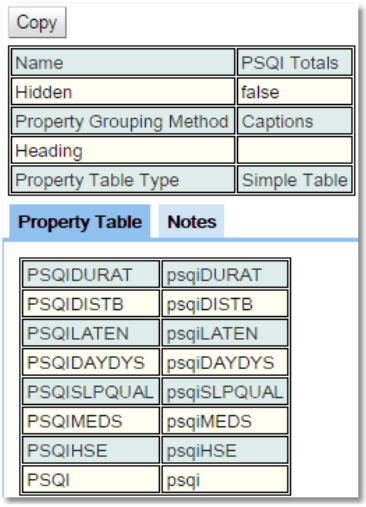
The search box can be used to find a *property*. Once the user starts typing a name into the search box, a list of possible *properties* will appear, as the user continues typing the possible list will reduce, until only one remains. The **Properties** branch containing that *property* will be expanded in the tree and the *property* will be highlighted.

If what has been typed does not match any possible *property* the search box will be outlined in red and the search text will also go red.

© 2010-2015 Exprodo Software

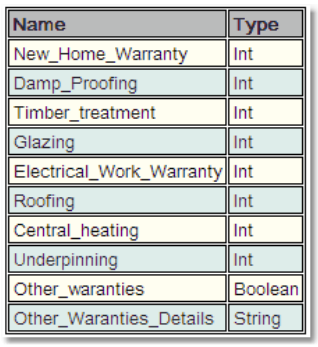
Information Pane

The **Information Pane** will have information about the item that is currently selected in the **Organisation Pane**. There are three types of items that can be selected
1. A **Property Group**.
2. A **Properties** branch
3. A *property*

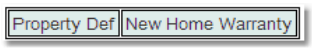


If a **Property Group** is selected then information about that group is displayed. Any *properties* associated with the **Property Group** are listed under the **Property Table** tab showing their **Name** and **Label**.
The **Notes** tab allows the user to define whether this **Property Group** contains a **Notes** field and/or a .
The **Copy** button allows the user put a copy of this **Property Group** into the buffer for later pasting.

Information	Description
Name	Name of the **Property Group**
Hidden	Whether the **Property Group** should be hidden from view.
Property Grouping Method	How the child **Property Groups** of this **Property Group** should be grouped/displayed. **Vertical Tabs**, **Vertical Tabs (Sorted by Name)**, **Horizontal Tabs** and **Captions**. To see examples of these layout types look at the Preview 416 image above.
Heading	Any heading to be displayed with this **Property Group**.
Property Table Type	How **Properties** belonging to this **Property Group** will be displayed. **Simple Table**, **Multi Column Table**, **Radio Button Table** and **Custom HTML**.



If a **Properties** branch to the tree is selected then the *properties* grouped here will be displayed as a list, showing their **Name** and **Type**.



If an individual *property* is selected then the *properties* **Label** will be viewed. This is so the user can see what will be written for this *property* in the layout.

© 2010-2015 Exprodo Software

Property Def	Preop Hernia
Show Full Detail	Show In Property Group Preop Hernia

If the *property* selected is of type *Biskit* then there will also be an option of how to display the *Biskit* information. Either as a **Show Simple**, (as a single element in a table) or **Show in Property Group** with a name assigned by the user (treating the *Biskit* as the first **Property Group**. If the *Biskit* has a layout defined it will be used, otherwise all the *properties* will be displayed as a simple table.)

Primary Key	298
Type	Biskit
Name	subjectStatus
Description	
Label	Subject Status
Tooltip	
Sort Order	0
Live	true
Min	
Max	
Attributes	☑ Visible ☑ Editable ☑ Persistent ☐ Null Allowed ☐ Required ☑ Visible In Biskit Detail ☑ Visible In Biskit List ☑ Visible In Collection Editor
Rows	
Columns	
Group	
Storage Mechanism	Static
Automated Property Type	
Biskit Def	Subject Status
Component	☐
Biskit Property Type	To One
Reference Deletion Option	No Action
Default Value	

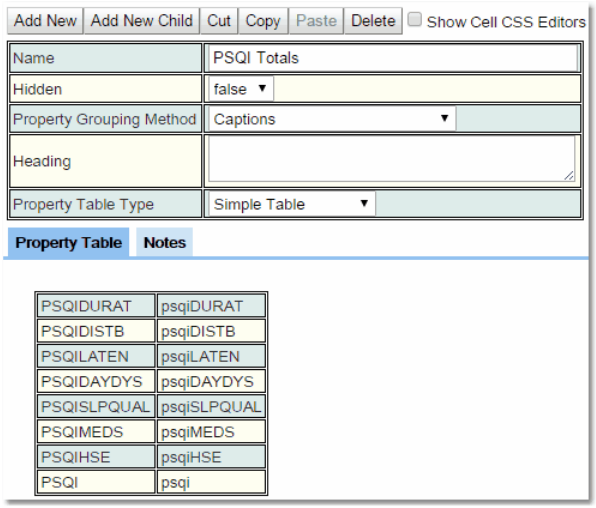
If a *property* is selected from the unassigned *properties* branch the information from the Bakery[351] about the *property* is displayed for information. This information can **NOT** be edited here even if the layout is in edit mode. To edit the details of a *property* the user will need to go to the Bakery[351].

© 2010-2015 Exprodo Software

Information Pane Edit Mode

On entering edit mode what is viewed in the **Information Pane** will vary depending on what is selected in the **Organisation Pane**.

If a **Property Group** is selected then the header information for the **Property Group** will be shown in an editable form. See the table[418] in the previous section for a description of the information displayed.



There will also be an edit toolbar.

Property Group Toolbar Item	Description
Add New	Adds a new **Property Group** at the same level as the currently selected **Property Group**.
Add New Child	Adds a new **Property Group** as a child of the currently selected **Property Group**.
Cut	Cuts the currently selected **Property Group**, removing it from its current place in the tree and puts it in the edit buffer overwriting any current contents.
Copy	Takes a copy of the currently selected **Property Group** and puts it in the edit buffer overwriting any current contents.
Paste	Pastes the current buffer contents as a child of the currently selected **Property Group**. When pasting a **Property Group** care needs to be taken if the **Property Group** has any *properties* associated with it. *Properties* can only appear once in a layout and the editor will remove *properties* that clash with the pasted *properties*.
Delete	Deletes the selected **Property Group**. Any *properties* associated with that **Property Group** will be moved to the **Unassigned Properties** branch.
Show Cell CSS Editors	This will display in each cell of the *properties* a little icon allowing the user to start up a CSS editor for that cell. For more on the CSS editor read the section on CSS Editors[413].

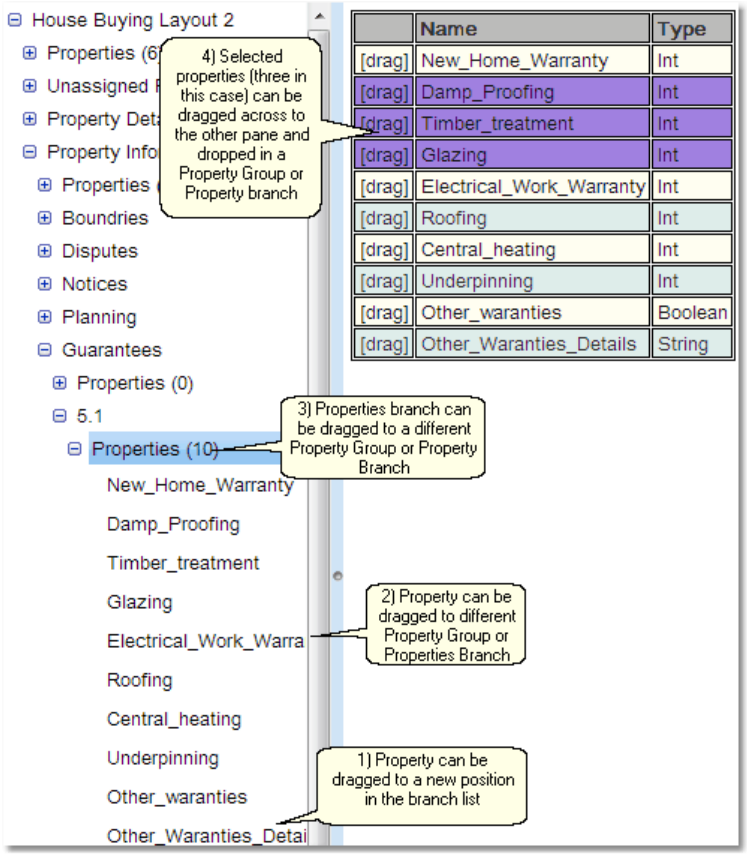
© 2010-2015 Exprodo Software

Moving Properties

If a **Properties** branch is selected then a list of *properties* will be shown in the **Information Pane**.
There are four ways to move *properties* within a combination of the **Organisation Pane** and the **Information Pane**

1. An individual *property* can be selected in the **Organisation Pane** and dragged to a new position in the list.
2. An individual *property* can be selected in the **Organisation Pane** and dragged and dropped into a different **Properties** branch.
3. A **Properties** branch can be selected in the **Organisation Pane** and dragged and dropped into a different **Properties** branch, all the *properties* in the dragged branch will then transfer to the new branch.
4. In the **Information Pane** using shift and ctrl, a number of *properties* can be selected from the viewed list and dragged across to a **Properties** branch in the **Organisation Pane**.

In all the above cases dropping the *properties* on a **Property Group** name will place the *properties* in the **Properties** branch for that **Property Group**.

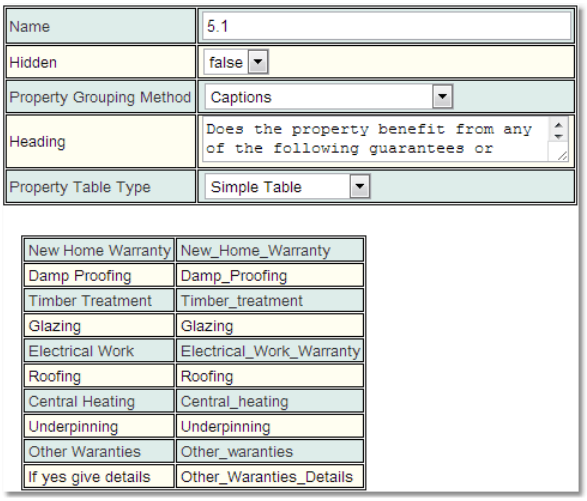


© 2010-2015 Exprodo Software

Property Table Types

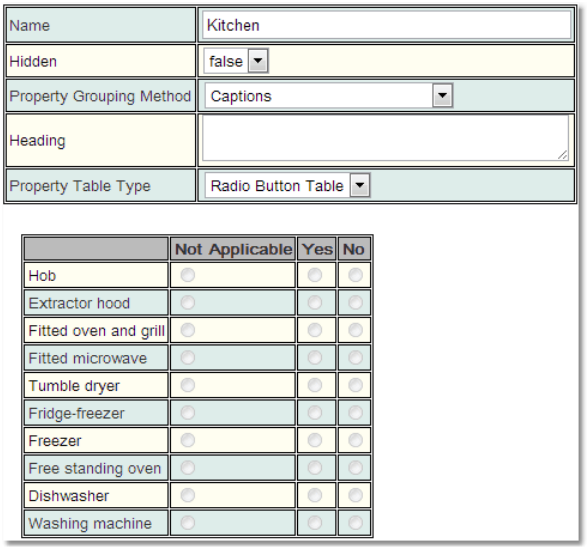
There are four different ways to display a list of *properties*.

1) Simple Table



Information in a **Simple Table** will be displayed with the *property* label and the *property* value.

2) Radio Button Table

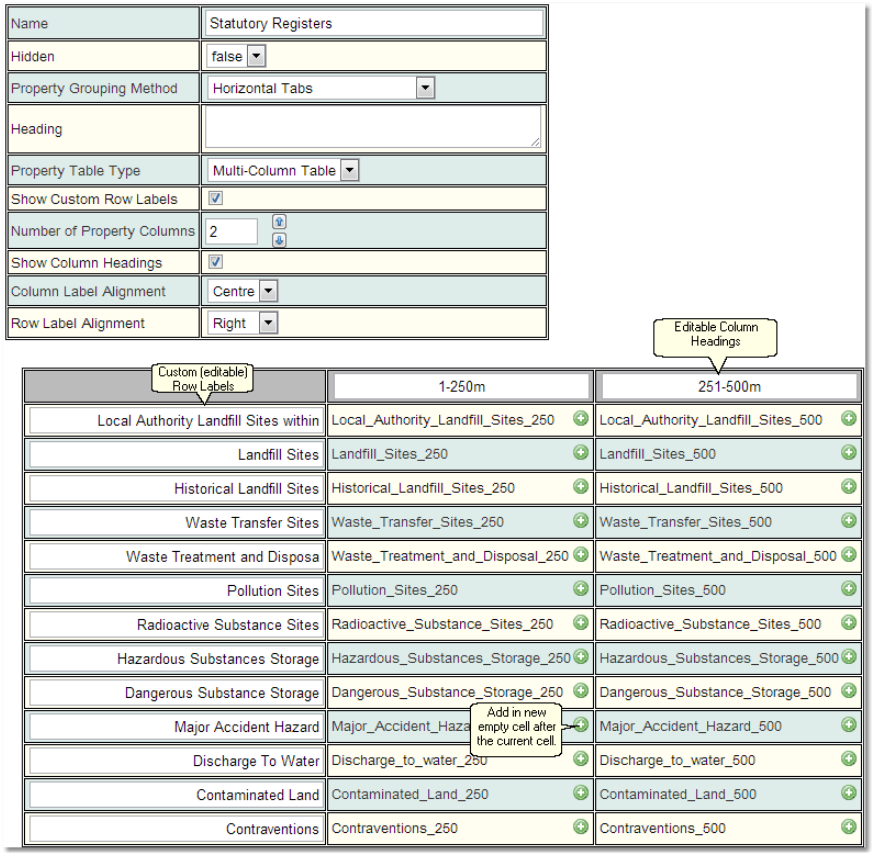


Information in a **Radio Button Table** is displayed showing the property labels and then a choice of values. This type of table is limited to the following

a. All *properties* must be of the same type.
b. They must be one of the following types:
   i. *Biskit* valued.
   ii. *Mapped Integer*[447].
   iii. *Mapped String*[447].

© 2010-2015 Exprodo Software

3) Multi-Column Table



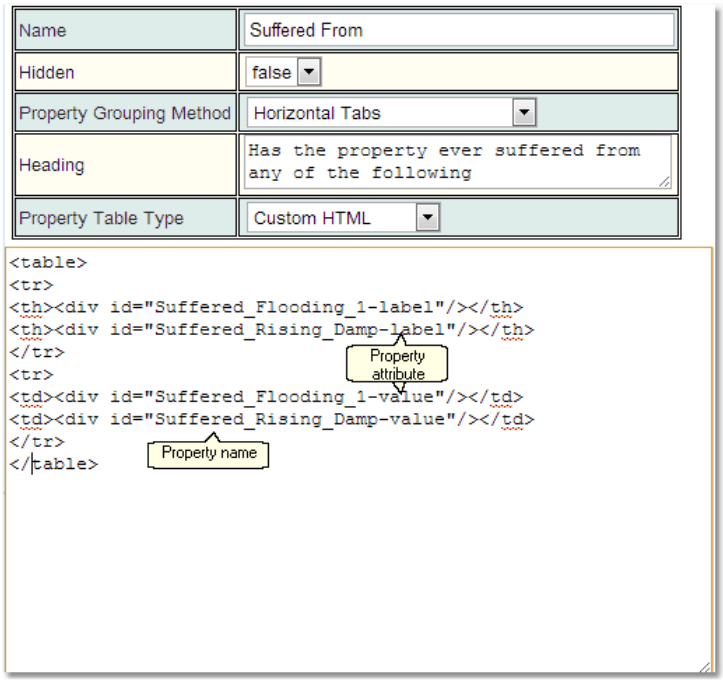
Information in a **Multi-Column Table** will be displayed with the *property* label and the *property* value. If multiple columns are being displayed only the labels of the *properties* required for the number of rows being used will be displayed.

Multi-Column Table	Description
Show Custom Row Labels	Allows the user to edit the row labels.
Number of Property Columns	The number of columns used to display the *properties*. The first *property* in the list will have its value displayed at the top of the first column, the next *property* value at the top of the next column and so on until the columns are filled up. The next available *property* value will then be displayed in the second row of the first column, *property* values will then fill in the second row before moving onto the next row and so on until all *property* values have been displayed
Show Column Headings	Displays column headings into which the user may place their own labels.
Column Label Alignment	Defines whether the column labels are left, centre or right justified.
Row Label Alignment	Defines whether the row labels are left, centre or right justified.
	Adds a new empty cell after the current cell. Useful for filling out tables which have a variable number of rows.

© 2010-2015 Exprodo Software

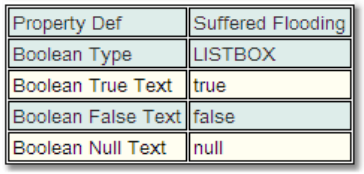
4) Custom HTML

Enable building of a table structure using HTML.



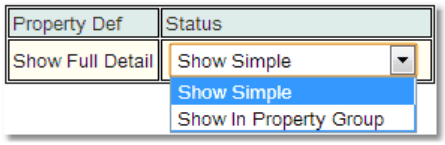
Other Display Options

Boolean *properties* have additional options, click on the *property* to get



The display type can be either a list box or a check box and the user can define the text to be used for **True**, **False** and **Null**.

Biskit *properties* have additional options, click on the *property* to get



The details of the *Biskit* can be displayed either as a simple table of all the *properties* (**Show Simple**) or in a separate **Property Group** (**Show in Property Group**) using the current layout type, **Horizontal**, **Vertical** or **Caption**. Also if a layout has been defined for the *Biskit* then the layout definition will be used for the *properties* of the *Biskit*.

© 2010-2015 Exprodo Software

Notes

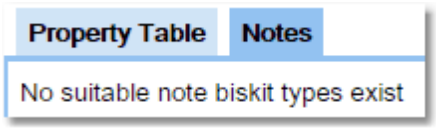
The user can add areas to the layout which enable the storing of notes. In order to do this a *BiskitDef* needs to be created to hold the note information. This *BiskitDef* must have at least two *properties*. Optionally the user can also define a *property* to hold the **Category** of note to be stored.

Property Type	Description
String	This *property* will hold the **Note** information. If the **StringPropertyType** is **HTML** then the note will support **Rich Text Format**.
Biskit	This property will define the parent *BiskitDef* that the **Note** is for. This *property* must be an **Automated Property Type**, set to **Create and Update**, the *Biskit Def* must be set to the parent *Biskit Type* and the **Biskit Property Type** must be set to **To One**.
Biskit	This property will define the **Category** *BiskitDef* that the **Note** is associated with. If used the **Category** is stored with the note and is used to determine which **Waterfall** the note will be displayed in.

Other *properties* may be added to the *BiskitDef* such as automated properties to hold the created and updated time, the creator, the updater and the version number as required.

There may be more than one **Note** *BiskitDef* if required. A single **Note** *BiskitDef* can be used to hold notes created in multiple places in the layout, or the user may define a different **Note** *BiskitDef* for each place notes are going to be added, or a combination of both methodologies.

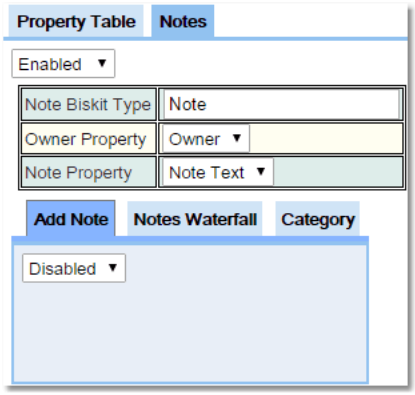
Once an approprite BiskitDef has been created the Notes tab will change from



to

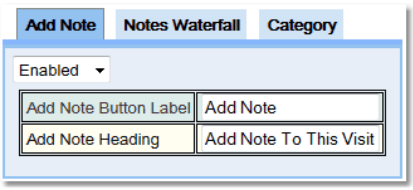
use the drop down to enable **Notes.**

Once enabled the user will see the following in the **Notes** tab.

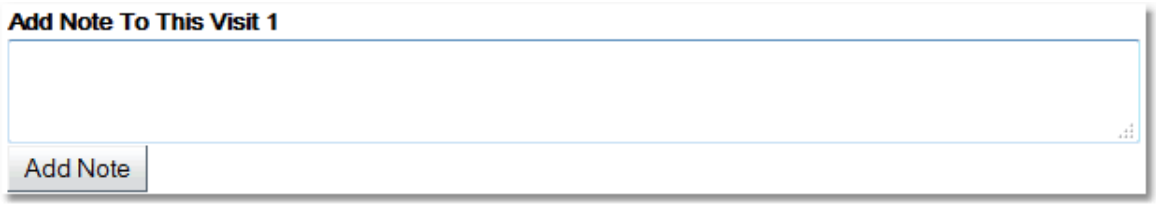


Note Biskit Type: This needs to be set to the *Biskit Type* that will hold the note information.
Owner Property: This needs to be set to the *property* that points to the owning *Biskit Type*. This is the *Biskit Type* that the notes can be attached to in the Layout Editor.
Note Property: This needs to be set to the *property* that will hold the note information.

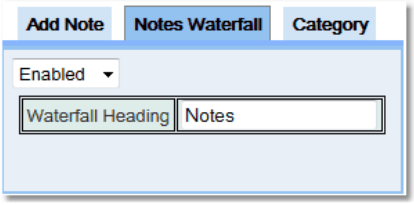
© 2010-2015 Exprodo Software



Add Note Button Label: This defines the string to be put on the button that will add the note to the database.
Add Note Heading: This defines the string that will be used as a heading to the add note area.
This information is used to determine how the add note area is defined. Below you can see an example of an add note area.



So far the area to write and add notes has been defined. Next the user needs to define an area to display all the notes that have been written to the DB, this is called a **Notes Waterfall**.

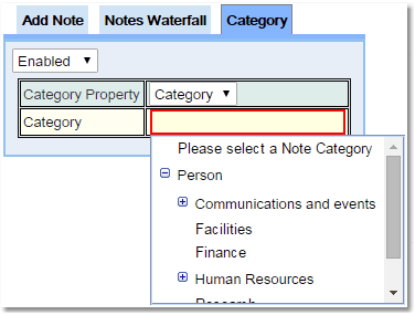


Once the **Waterfall** is enabled the user can define what heading will appear above the **Waterfall** in the layout.

Finally notes can be stored under a number of **Category**'s. In order to set up the **Category**'s a *hierarchical BiskitDef*[402] needs to be defined, this is where a parent can have many children of the same type of *BiskitDef* and a child will have only one parent. The minmum properties would be:

Property Type	Description
String	This *property* will hold the **Category** name.
Set	This property will point to the children's *BiskitDef* as a Set. These will be of the same *Biskit Type* as the parent.
Biskit	This property will point to the parents *BiskitDef*. This will be of the same *Biskit Type* as the child.

© 2010-2015 Exprodo Software



Once the **Category** is enabled the use will need to set the **Category** *property*. This is the property on the **Note** *BiskitDef* that points to the **Category** *BiskitDef*.

Then the user will need to define which of the **Category**'s the current **Notes** section will save **Notes** as. Any **Waterfall** enabled in this **Notes** section, will only display **Notes** of this **Category.** If no **Category** is enabled then the **Waterfall** section will display all **Notes** no matter what their **Category**. An **Add Note** can define a **Category** to create a **Note** under in one place in the **Layout** and a **Notes Waterfall** can be enabled with that **Category** defined somewhere else in the **Layout** if required.
Permissions [448] may be assigned to **Category**'s to stop/ allow viewing/editing of notes. A *Permission* assigned to a **Category** in a hierarchy will be inherited by all its children.

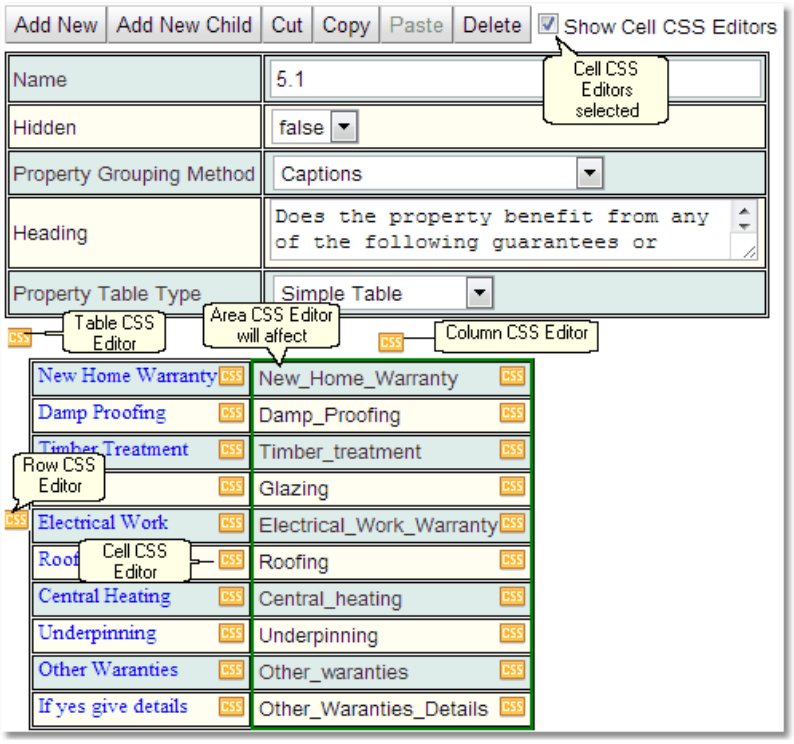
© 2010-2015 Exprodo Software

CSS Rule Editors

The user can define their own CSS information for each Table, Row, Column or Cell using the **CSS Editors**. These editors are displayed as small icons with CSS written on them. In order to bring up the icons:

1. For a row, move the cursor to just to the left of the start of the row.
2. For a column, move the cursor to be just above the column.
3. For a table, move the cursor to be just outside the top left hand corner of the table.
4. For a cell, check the **Show Cell CSS Editors** box in the **Property Group Toolbar**.

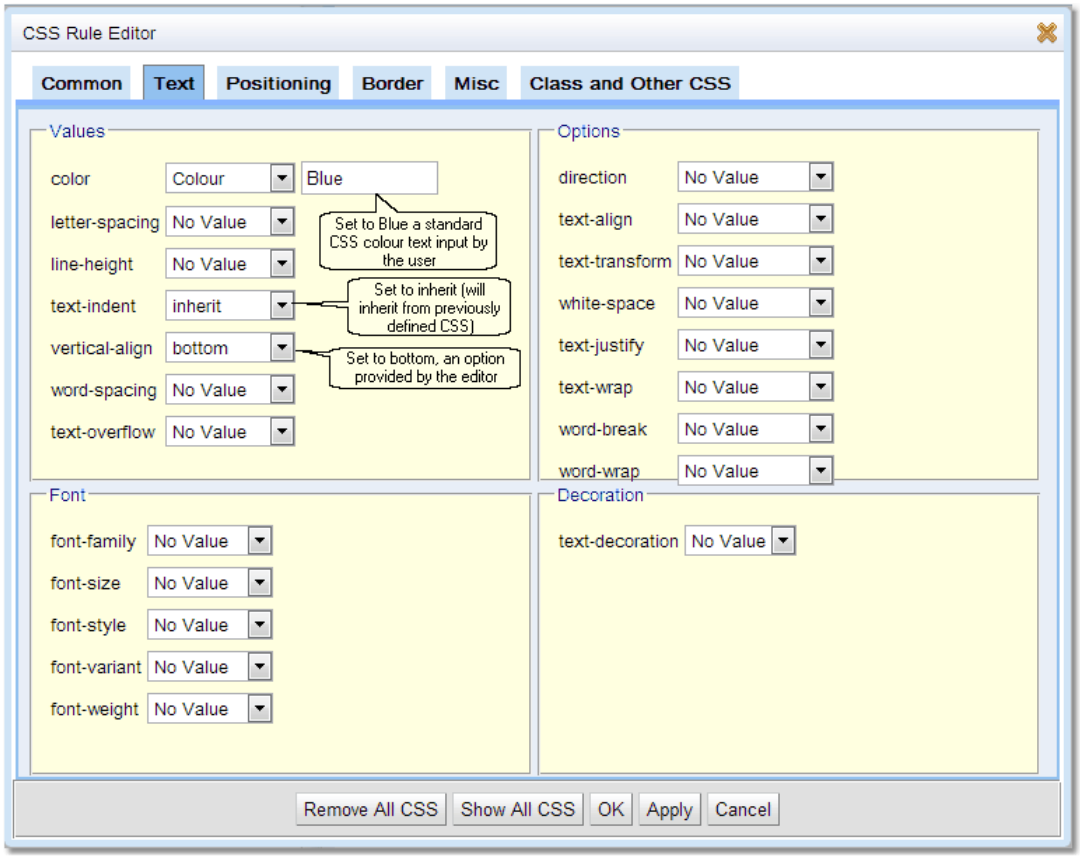
When selecting a **CSS Rule Editor** the area to be affected will have a green border.



In the example above the first column has had the text colour set to Blue, the font-family set to Times New Roman and the font-size set to small.

© 2010-2015 Exprodo Software

When the **CSS Rule Editor** is opened up:



The possible CSS values are organised into different tabs to help with finding the required CSS value. When the drop down button is pressed for a value the user will have a choice of options. Some will require the user to input the appropriate CSS string (see Colour above). The user can also specify inherit which means the CSS will be inherited from previously defined CSS by the user, this could be in Global Preferences[329] or elsewhere in **Calpendo** where CSS can be defined.

For a complete description of the standard toolbar buttons read the Toolbar Button Standard Definition[96] chapter.

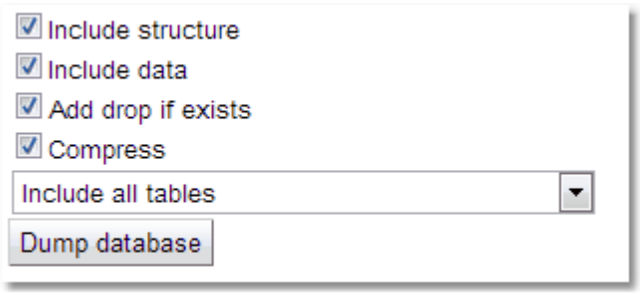
CSS Editor Toolbar Item	Description
Remove All CSS	Removes all CSS that the user has applied using the editor to this particular object.
Show All CSS	Shows in a pop up window the CSS that has been defined for this particular object.

For more information and help on using CSS and HTML a site we recommend is http://www.w3schools.com/

© 2010-2015 Exprodo Software

6.19 Backing Up The Database

To take a backup of the database, use the **Database Dump** page, which is by default on the menu as **Admin-->Database Dump**.



Before asking for the backup, there are the following options:

Option	Description	
Include structure	If ticked, the backup will contain SQL statements that will create the database tables.	
Include data	If ticked, the backup will contain SQL statements containing the data from the database.	
Add drop if exists	If ticked, the before each database table is created, a statement is added to drop that table if it already exists.	
Compress	If ticked, the backup will be compressed before being downloaded.	
Include all tables	This drop down allows the choice of one of the following options:	
	Option	**Description**
	Include all Tables	Dumps all tables
	Include all tables except history and statistics	Dumps all tables except history and statistics
	Select Tables	Provides a simple choice of tables to be dumped.
	Use Custom per-table settings	See below for complete description

When all the settings are as required, just press the **Dump database** button to initiate the backup. When the backup is complete, the browser should download the backup file. **Calpendo** limits how often a database backup may be taken, by setting **Minimum time between database backups** on the General tab of the Global Preferences[328].

© 2010-2015 Exprodo Software

Use Custom Per-Table Settings

If the **Use Custom per-table settings** option is chosen, then **Calpendo** will get a list of all the tables in the database and allow the way each table is backed up to be customised.

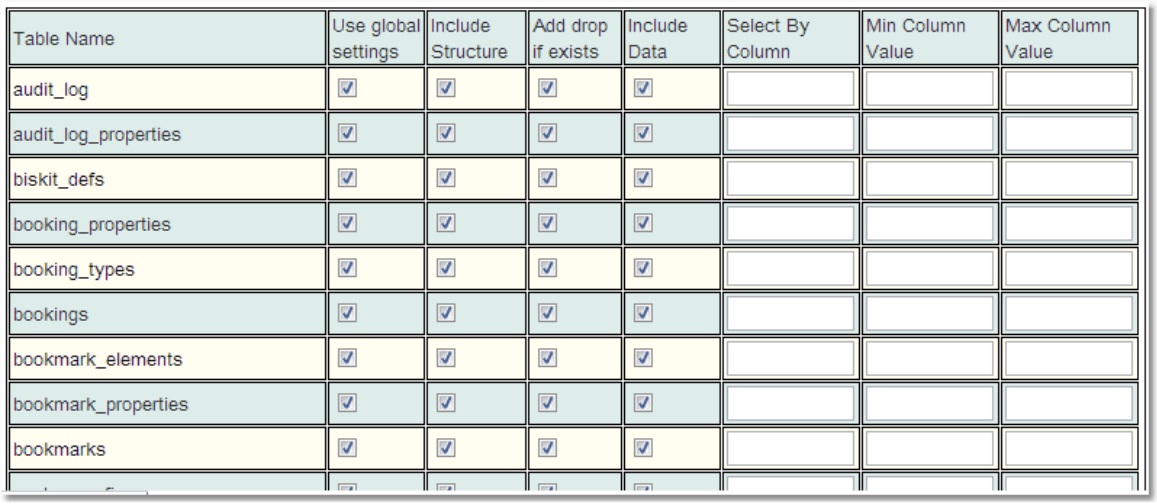


Table Name	Use global settings	Include Structure	Add drop if exists	Include Data	Select By Column	Min Column Value	Max Column Value
audit_log	☑	☑	☑	☑			
audit_log_properties	☑	☑	☑	☑			
biskit_defs	☑	☑	☑	☑			
booking_properties	☑	☑	☑	☑			
booking_types	☑	☑	☑	☑			
bookings	☑	☑	☑	☑			
bookmark_elements	☑	☑	☑	☑			
bookmark_properties	☑	☑	☑	☑			
bookmarks	☑	☑	☑	☑			

For each table, there are the following options:

Option	Description
Use global settings	If ticked, then the other options for this table will be ignored.
Include structure	Allows the global setting of **Include structure** for this table to be overridden.
Add drop if exists	Allows the global setting of **Add drop if exists** for this table.
Include data	Used to override the global setting of **Include data** for this table to be overridden.
Select by column	If partial data is to be downloaded, then the user can nominate a column of the table and specify minimum and/or maximum values of that column that should be included in the download.
Min Column Value	If the user wants to download partial data and specifies **Select by column**, then the **Min column** value is the minimum value of the selected column that should be included in the download.
Max Column Value	If the user wants to download partial data and specifies **Select by column**, then the **Max column value** is the maximum value of the selected column that should be included in the download.

© 2010-2015 Exprodo Software

6.20 Creating A Read Only Copy

If **Calpendo** is installed on your own servers, it may be necessary to create a backup instance that runs against a copy of the database. Normally such a copy would want to be configured to be read-only so that people can connect to it to access information, but cannot change anything.

There are three ways to place your **Calpendo** into a read-only mode:

1. Set up the access rights for the database user so that it does not have insert *Permission*[448] to the **exprodo_events** table. A *system event*[449] is created at boot time, and if the insert fails, this is used as a trigger to assume a read-only mode.
2. Set the read-only mode[339] setting in the Global Preferences'[328] General[339] tab.
3. Change the read-only mode by directly modifying the database. Here is an example of the SQL you would use if you are using MySQL to store the **Calpendo** database:
4.

```
update config_properties set boolval='T' where name='readOnlyMode';
```

To manually modify the database to disable read-only mode that is trigged by setting the *global preferences*, run this SQL:

```
update config_properties set boolval='F' where name='readOnlyMode';
```

Requirements For A Read-Only Database

If you create a database and configure **Calpendo**'s database user so that it has no insert or update *Permissions*, then **Calpendo** will not work properly. To make it work, you must at least have the following *Permissions* (this is expressed in MySQL code):

```
GRANT SELECT ON `calpendo`.* TO 'calpendo_ro'@'localhost' identified by
'thepassword';
GRANT INSERT, UPDATE ON `calpendo`.`audit_log_properties` TO
'calpendo_ro'@'localhost';
GRANT INSERT, UPDATE ON `calpendo`.`sessions` TO
'calpendo_ro'@'localhost';
GRANT INSERT, UPDATE ON `calpendo`.`user_settings` TO
'calpendo_ro'@'localhost';
GRANT INSERT, UPDATE ON `calpendo`.`audit_log` TO
'calpendo_ro'@'localhost';
GRANT INSERT, UPDATE ON `calpendo`.`user_resource_info` TO
'calpendo_ro'@'localhost';
GRANT INSERT, UPDATE ON `calpendo`.`login_attempts` TO
'calpendo_ro'@'localhost';
```

where **calpendo_ro** is the name of the database user.

© 2010-2015 Exprodo Software

Consequences of Putting Calpendo Into Read-Only Mode

In read-only mode, the following apply:

- Reminders emails are disabled. This is because **Calpendo** cannot track which reminders have or have not been sent.
- Automatic Emails do not run
- When using the *Bookings Calendar*[446], it does not offer the ability to edit anything
- Normally, when a *repeat*[449] *booking*[446] or *repeat Time Template*[449] passes a due time, a new instance of the *booking* or *Time Template* is created so that **Calpendo** keeps a record of what *bookings* or *Time Templates* were in effect. It means that all past records should be accurate. In read-only mode, no such modification takes place. This means that the record of the past is not the same as it would be for a read-write **Calpendo**.

© 2010-2015 Exprodo Software

Part VII

7 Calpendo Installation Guide

If your **Calpendo** is to be hosted on **calpendo.com**, then you will not need to read these instructions as it will be installed for you. You only need to read this installation guide if you are going to install **Calpendo** on your own server.

Hardware and Software Requirements

Operating system

- Windows
- MacOS
- Linux

Server Software

- Java 7 minimum (when we add support for HTTP/2, this will require Java 8)
- MySQL 5.1 minimum (or MariaDB equivalent)
- Tomcat 7 minimum
- Apache HTTP Server (tested on 2.2 and 2.4, although any version should work)

Server Hardware

- RAM 4GB minimum, 8GB recommended
- Hard Disk 10GB minimum, but requirements depend on usage, particularly if you configure the system to store file uploads.
- Recent & fast CPU recommended
- Both 32 bit and 64 bit supported
- You can use a virtualised server if required

Client Software

- Chrome - any version, but recent versions preferred
- FireFox - any version, but recent versions preferred
- Safari - any version, but recent versions preferred
- IE 9 or later (IE8 mostly works but is extremely slow)

Expected Future Changes

We expect the following changes to occur to our requirements:

- Tomcat: Some time during 2015 or 2016, we expect to drop the requirement for the use of Tomcat. We expect to replace Tomcat with Jetty, which we will bundle with our software.
- Java: Java 6 and Java 7 are both no longer receiving public updates from Oracle. Further, when we add support for HTTP/2 (which will coincide with dropping the requirement for Tomcat), HTTP/2 will only work with Java 8, and so we may require the use of Java 8 at that time.

© 2010-2015 Exprodo Software

Requirements

You should have received a file archive that contains the following files:

```
calpendo-x.y.z.sql
Calpendo/
INSTALL/apache/ssl-www.calpendo.com
INSTALL/apache/www.calpendo.com
INSTALL/apache_with_mod_jk/ssl-www.calpendo.com
INSTALL/apache_with_mod_jk/mod_jk.conf
INSTALL/apache_with_mod_jk/singleTomcatInstance/ssl-www.calpendo.com
INSTALL/apache_with_mod_jk/singleTomcatInstance/workers.properties
INSTALL/apache_with_mod_jk/singleTomcatInstance/www.calpendo.com
INSTALL/apache_with_mod_jk/workers.properties
INSTALL/apache_with_mod_jk/www.calpendo.com
INSTALL/hibernate.cfg.xml
INSTALL/README.txt
INSTALL/tomcat6/conf/server.xml
INSTALL/tomcat6/conf/tomcat-users.xml
```

The **Calpendo** directory contains the program itself, and `calpendo-x.y.z.sql` contains a blank **MySQL** database. The other files are examples for configuring **Apache Tomcat 7** and **Apache** web server, described further below.

Installation Overview

You will need:

- A database server, eg **MySQL**, though many other database servers will work ok
- A servlet engine, eg **Tomcat 7**, **Jetty** etc
- **Java JDK 7** (or possibly just a **JRE**, depending on the version of your servlet engine)
- An **Apache HTTP** server may be required, or else you can serve directly from the servlet engine. The config files are use **Apache HTTP** server as a front end to **Tomcat**.
- A database. If using **MySQL**, then you can load the `calpendo-x.y.z.sql` provided. For any other database, **Calpendo** can be made to generate the necessary tables itself (instructions for this are not here)

Once those things are in place, you need to:

- Copy the **Calpendo** directory into the servlet engine's `webapps` directory
- Change a configuration file so it knows how to connect to the database
- Configure the servlet engine and **Apache**
- Start the servlet engine and **Apache**
- Connect to the server from a browser to make sure it is accepting connections.

© 2010-2015 Exprodo Software

Install MySQL

Any version that supports **InnoDB** tables should do, but the database dump that's included in this distribution comes from **MySQL 5.5.x**. **MySQL** documentation should be consulted if loading this into any other version.

Create The Database And A Mysql User

Open a connection to **MySQL** with:

```
mysql -u root -p
```

and then at the `mysql` prompt:

```
mysql> create database calpendo;
mysql> grant all privileges on calpendo.* to calpendo@localhost
    -> identified by 'somepassword';
mysql> exit;
```

You may choose a different name for the database and/or user if you wish. Later on, you will configure **Calpendo** so it can locate the database and log in - you will just put the connection information into a URL (in file `hibernate.cfg.xml`).

Load The Sample Database

You should have a file called `calpendo-x.y.z.sql` that you can load:

```
mysql -u calpendo --password=somepassword calpendo < calpendo-x.y.z.sql
```

Install A Java JDK

Follow the installation instructions for the **JDK** for your platform. This needs to be at least **JDK 7**. Note that the **JDK** is required, not just the **JRE** (this is a requirement for **Apache Tomcat**, the servlet server, rather than **Calpendo** itself).

© 2010-2015 Exprodo Software

Install A Java Servlet Server

This should be **Apache Tomcat**, or **Jetty** or some other server that supports the **Servlet 2.5** specification. **Calpendo** has only been tested with **Tomcat 7**, and so this is the preferred server. A future **Calpendo** release is likely to use **Jetty** instead (and also include a much easier installation and upgrade mechanism). The rest of this section assumes you are using **Tomcat** version **7**.

You should read and follow the standard documentation for **Tomcat** to install it. Once installed, you will need to edit two files in **Tomcat's** `conf` directory, `server.xml` and `tomcat-users.xml`. You should find samples of each of these with the **Calpendo** distribution. The important parts of these are:

`tomcat-users` This needs to contain a `<user...>` line so that somebody can connect to **Tomcat's** web management pages. The user name and password can be anything and will only ever be used by a human. You may want more than one entry here, depending on who will do the administration.

`server.xml` The sample provided defines a `<Connector ...>` entry on port 8080 that allows users to connect directly to **Tomcat**. A standard **Tomcat** `server.xml` would contain this entry, although I've added the necessary config so that it supports compression. If all access is to be made through **Apache**, then this connector on port 8080 could potentially be turned off. Leave it on for now so you can test the installation before we bring **Apache** into the equation.

I've also added a `<Connector...>` on port 8009 with protocol **AJP/1.3**. This is the means by which **Apache** talks to **Tomcat**.

Once done, check that you can connect to **Tomcat** and access its management pages. Go to the web page:

```
http://servername:8080/
```

where `servername` is the name of the server you installed **Tomcat** on. There should be a link on there to the management pages so you can ask **Tomcat** to start **Calpendo** (after **Calpendo** is installed).

© 2010-2015 Exprodo Software

Install Calpendo

Calpendo is delivered as a directory in war file format. However, there's a file inside the directory (`WEB-INF/classes/hibernate.cfg.xml`) that you will need to modify to tell **Calpendo** how to connect to the database.

The root directory in the archive is called `Calpendo`. You should put this directory into **Tomcat's** `webapps` directory. Then edit the file

```
webapps/Calpendo/WEB-INF/classes/hibernate.cfg.xml
```

The entries you need to configure are:

```
<property name="hibernate.connection.username">
   calpendo
</property>
<property name="hibernate.connection.password">
   somepassword
</property>
<property name="hibernate.connection.url">
   jdbc:mysql://localhost/calpendo
</property>
<property name="hibernate.connection.driver_class">
   org.mariadb.jdbc.Driver
</property>
<property name="hibernate.dialect">
   com.springsolutions.exprodo.core.server.persistence.ExprodoMySQL5Inno
   DBDialect
</property>
```

You need to change the username, password and url to match the user and database name you created earlier.

You should now ask **Tomcat** to start **Calpendo** (in case it isn't done automatically). Go to the web management page, linked from:

```
http://servername:8080/
```

and see if **Calpendo** has started, and if not, ask for it to be started.

Once this is done, then you should test to see whether **Calpendo** is working with **Tomcat**. Open a web browser at this url:

```
http://servername:8080/Calpendo/com.springsolutions.calpendo.Calpendo/
program.html
```

All being well, you will see a **Calpendo** login screen. If you don't, then you'll need to check to see what errors are on screen or in the **Tomcat** logs.

If you do see the login screen, then try to log in with **username** `root` and a blank **password**.

© 2010-2015 Exprodo Software

Install Apache HTTP Server

Follow the **Apache** documentation as appropriate for your operating system in order to install the **Apache HTTP** server. Once installed, you should make sure your **Apache** configuration loads modules **mod_proxy**, **mod_proxy_http**, **mod_rewrite**, **mod_deflate** and **mod_ssl**. The location of your modules may vary from that shown below, so you may need to alter this sample configuration:

```
LoadModule proxy_module /usr/lib/apache2/modules/mod_proxy.so
LoadModule proxy_http_module /usr/lib/apache2/modules/mod_proxy_http.so
LoadModule rewrite_module /usr/lib/apache2/modules/mod_rewrite.so
LoadModule deflate_module /usr/lib/apache2/modules/mod_deflate.so
LoadModule ssl_module /usr/lib/apache2/modules/mod_ssl.so
```

For **mod_deflate**, you will need to add config like this:

```
<IfModule mod_deflate.c>
    AddOutputFilterByType DEFLATE text/html text/plain text/xml text/css
text/javascript
</IfModule>
```

This allows **Apache** to `gzip` the files it serves. This has a marked impact on the size of **Calpendo javascript** files in particular.

You will now need to configure the **Calpendo** aspects of **Apache**. You will have your own requirements here, but here is the configuration you will need to provide access to **Calpendo** from a URL of "*https://yourServer/Calpendo*".

```
RewriteEngine On
RewriteRule ^/Calpendo/Service(.*) /Calpendo/com.springsolutions.calpendo.Calpendo
RewriteRule ^/Calpendo/program.html(.*) /Calpendo/com.springsolutions.calpendo.Cal
RewriteRule ^/Calpendo/$ /Calpendo/com.springsolutions.calpendo.Calpendo/program.h
RewriteRule ^/Calpendo/(.*) /Calpendo/com.springsolutions.calpendo.Calpendo/$1 [PT
ProxyRequests Off
<Proxy *>
    Order allow,deny
    Allow from all
</Proxy>
ProxyPass /Calpendo http://localhost:8080/Calpendo
ProxyPassReverse /Calpendo http://localhost:8080/Calpendo
```

The above configuration should be placed inside a **VirtualHost** declaration. If you want to provide both **SSL** and **non-SSL** access to **Calpendo**, then you should place the above configuration in both the **SSL** and **non-SSL VirtualHost** areas. At this point, accessing **Calpendo** from "*https://yourServer/Calpendo*" should work, possibly unless you are using an operating system that uses **SELinux**, as described below.

Note that previous versions of this installation guide differed in two ways from the above:

- Previously, it was suggested that you should use **mod_jk** to connect **Apache HTTP** server with your servlet container (such as **Apache Tomcat**). This is no longer recommended. In particular, **mod_jk** isn't available for recent versions of **Red Hat Linux** and **CentOS**. That is why the above recommended **Apache** configuration uses **mod_proxy_http**.

© 2010-2015 Exprodo Software

- It used to be a requirement that **http** access was required, at least from `localhost`, even when you only wanted **https** access to be remotely accessible. This requirement no longer exists so you only need to configure the **SSL VirtualHost** with the above directives if you only want to provide **SSL** access.

SELinux Settings

If you are running a version of **Linux** that uses SELinux, then you may find that your **Apache HTTP** server is prevented from opening further **http** connections, thereby stopping the above from working. You can check whether you will need to do this by running this command:

```
getsebool httpd_can_network_connect
```

If this returns zero or off, then you will need to change this setting by running this command:

```
setsebool -P httpd_can_network_connect 1
```

You can read about the `httpd_can_network_connect` setting on the httpd_selinux man page. If your `httpd_can_network_connect` setting is zero or off, then the result will be an error in your **Apache** logs similar to this:

```
(13)Permission denied: proxy: HTTP: attempt to connect to 127.0.0.1:8080 (localhos
```

Rebooting

Finally, you will need to arrange for **MySQL**, **Tomcat** and **Apache** to be started each time your operating system boots. You should follow the standard method for your platform. For **Windows**, this means they should all have been installed as services.

Advanced Apache Configuration

Sometimes, you may have an external-facing **Apache HTTP** server through which all of your traffic flows (eg on a firewall) and then internally you may have another **Apache HTTP** server that has the **mod_jk** module installed. In this situation, with two **Apache HTTP** servers and one **Tomcat** server, there is some extra configuration to do. Without any extra configuration, **Tomcat** will always be given the IP address of the firewall **Apache HTTP** server as the source of each request instead of that of the user's computer. The problem with such a situation is that the login attempts that **Calpendo** will record will all show the firewall as the source of every attempt. For defence against any attack, it would be better to change this.

You can set up forwarding of **HTTP** requests from the firewall to the inner **Apache HTTP** server in two ways: by configuring a reverse proxy with the **ProxyPass** directive, or with **mod_rewrite** and the [P] directive.

If you use **ProxyPass**, then you can make a configuration change on the firewall **Apache HTTP** server by adding the directive

```
ProxyPreserveHost On
```

to the appropriate part of the firewall's **Apache HTTP** server configuration.

This situation is described here:

```
http://tomcat.apache.org/connectors-doc/generic_howto/proxy.html
```

You may be able to achieve the desired effect with **SetEnv** or **SetEnvIf** directives instead of the above (but I have not tested this).

© 2010-2015 Exprodo Software

Once you have done this, you can check whether this is working as expected in one of two ways. First, you can look at entries that appear in the **login_attempts** table whenever somebody tries to login to **Calpendo**. Secondly, you can check from **Calpendo** by using either the Search[108] page or the Data Explorer[127] page.

Upgrading Calpendo

Upgrades follow the following path:

1. Stop **Tomcat**
2. Take a backup of your **MySQL** database:

   ```
   mysqldump -u root --password=somepassword calpendo >
   calpendo.saved.sql
   ```

3. Apply any database upgrade script provided for your release. For a release of version "a.b.c", this would mean:

   ```
   mysql -u root --password=somepassword calpendo < update-a.b.c.sql
   ```

4. Replace the **Calpendo** directory with the contents of the new **Calpendo** directory. Note that you must ensure that the `hibernate.cfg.xml` file that you customised when you followed the above installation instructions is not overwritten by the new release. New releases normally do not include a copy of `hibernate.cfg.xml` file.
5. Restart **Tomcat**
6. Open a web browser to confirm that **Calpendo** is accepting connections

Rolling Back a Failure Database Upgrade

If you should want to roll back the database because an upgrade failed in some way, then you should *not* load you back-up over the top of the upgraded database. That's because there could be tables for the upgrade that remain in place, and so it might cause problems with doing the upgrade again (since it may assume the new tables do not exist).

The recommended way to roll back the database after an upgrade is to drop the database, recreate it, and then load the backup.

© 2010-2015 Exprodo Software

Part VIII

8 Glossary

Term	Definition
Abdicated	The process of indicating that you will not use a booking[446] slot allocated to you by a Time Template[450], so that it can be used by other people.
Acceptable	A Time Template[450] is acceptable when bookings[446] matching the Time Template[450] should be allowed, and should be given the status of Requested.
Action	See Permission Action[448].
Applies To	Defines the list of users that are will affected. This list is overruled by the Does Not Apply To[447] list. ie. If a user is in both lists then the Does Not Apply To[447] is used.
Approved Booking	A booking[446] whose status is approved.
Approved User	A user that has been approved for access to the system.
Approved Project	A project[448] that has been approved for use by an administrator.
Automatic Approval	A Time Template[450] has a state of automatic approval when bookings[446] matching the Time Template[450] should be allowed, and automatically given the status of Approved.
Automatic Denial	A Time Template[450] has a state of automatic denial when bookings[446] matching the Time Template[450] should not be allowed.
Bakery	An editor which allows the user to create, delete and update BiskitDefs[446] and their properties[448]. The Bakery also allows editing of Mapped Strings[447], Mapped Integers[447] and Units[450].
Best Possible	A booking status[447] that indicates that **Calpendo** should create an approved booking[446] if possible, and a requested booking[449] otherwise.
Biskit	A Biskit is an object stored in the database that has a custom type. It will have whatever properties[448] are specified by its BiskitDef[446].
BiskitDef	Defines the properties[448] associated with a particular type of Biskit[446] and how those properties[448] should be stored in the database.
Biskit Type	Each Biskit[446] has a type. Biskits[446] with the same type, have the same property[448] definitions. See BiskitDef[446].
Bit Set	A Bit Set is a type of integer property[448] that results in a drop-down that can take multiple values.
Booker	The user who has made a particular booking[446].
Booking	A reservation of a resource[449] at a particular time. Bookings may also be repeatable[449].
Booking Approval Process	The process by which a booking[446] becomes approved.
Bookings Calendar	A page which provides a graphical view of bookings[446] by day, week or month.
Booking Rule	A restriction relating to which bookings[446] can be made.
Booking Rule Type	The different types of booking[446] regulations. E.g. Simple, Double, Advanced, Total Time, Number of Bookings etc.

© 2010-2015 Exprodo Software

Booking Rule Validator	A tab in the Booking Rule Editor that provides a means of testing Booking ules 446.
Booking Status	The current condition of the booking 446. There are four possible statuses Requested, Approved, Denied or Cancelled.
Booking Type	A user-defined Booking 446 taxonomy.
Bookmark	A collection of resources 449 to be displayed in the Bookings Calendar 446. Each user can create their own personal bookmarks, and there may also be system wide bookmarks.
Condition	Allows the user to specify under what circumstances an action such as generating an email, granting or refusing permission 448 etc. will occur. Conditions work logically together using information from the database to reach a True/False result.
Data Explorer	A page that shows Biskits. 446
Data Type	See Biskit Type 446.
Default Booking Status	A globally set up preference that specifies the status of a booking 446 if no Time Template 450 is available.
Does Not Apply To	Defines the list of users that an operation (Rule 446, Permission 448 etc) will not affect. This list overrules the Applies To 446 list. ie. If a user is in both lists then the Does Not Apply To is used.
Event	An **Exprodo SDM** concept which defines a when a task 449 needs to occur.
FAQ	Frequently Asked Questions, this page lists the FAQ's for the facility.
Far Term Time Time Template	Time Templates 450 allow the option of having a different Time Template used beyond a certain time period. This is usually to restrict bookings 446 beyond this time period, this is known as the Far Term Time Template.
Global Preferences	Specifies settings that change the way **Calpendo** operates.
Group Report	A report that lets you break down data by any number of properties 448.
History	The audit log kept by **Calpendo** that shows what has been changed, when and by whom.
Layout Editor	The Layout Editor allows the user to define how the properties 448 of a Biskit 446 will be displayed.
List Report	A report that shows a list of Biskits 446.
Mapped Integer	A Mapped Integer relates text labels to integer values. An integer property on a Biskit 446 may use a Mapped Integer so that a user would see a drop-down with each of the text labels when they edit it. The related integer value is stored in the database rather than the text. See also Bit Set 446.
Mapped String	Similar in concept to a Mapped Integer 447, but instead of storing an integer in the database, a text value is stored instead. This means that a Mapped String is an association of one set of strings to another set of strings so that you see a drop-down of text values, but the value stored in the database does not have to be the same as the displayed value.

© 2010-2015 Exprodo Software

Meta-Property	1) Attributes that define how a Biskit[446] or a Biskit's[446] properties[448] work. 2) A type of property used in Conditions[447] when you want to specify a relation[449] about who performs an action or when.
Network Metrics	A record of how long communications between web browsers and the **Calpendo** server take.
Nobody	A special user used for Permissions[448] to control what information may be written into emails.
Near Term Time Template	Time Templates[450] allow the option of having a different Time Template known as the near term Time Template used when getting close to a particular booking[446] time. This allows unused booking[446] slots to be used.
Owner	The user that owns a booking[446], bookmark[447] or project[448]. Bookings[446] may be owned by users other than the booker[446].
Permission	How the administrator determines which user may have access to what information and under which circumstances. Permissions may have many layers allowing fine grained control over who can perform what action[446] on which Biskit[446].
Permission Action	An operation that can be performed on a Biskit[446]. The most common actions are Create, Read, Update and Delete.
Project	A project allows an administrator to link together users, resources[449] and costs as well as other customised information to enable them to organise who can make bookings[446] for particular resources and how they will be charged.
Project Approval Process	The process by which new projects[448] become approved.
Project Code	A unique string used to define a project[448].
Project Group	A way of grouping similar projects[448] together for use in Rules[446], Permissions[448], Time Templates[450] or Automatic Emails. A project[448] may belong to many groups. See also Project Type[448].
Project Resource Settings	The list of resources[449] that a project[448] may use, including information that may be added such as the price or booking[446] duration for each resource[449].
Project Service Settings	The list of services[449] that a project[448] may use, including information that may be added such as the price.
Project Status	The current status of a project. This can be **Requested**, **Approved**, **Denied**, **Unbookable** or **Terminated**.
Project Template	A project[448] that would normally have a status of Unbookable, and is used to set default values for newly created projects[448].
Project Type	A way of grouping similar projects[448] together. A project has only one type. See also Project Group[448].
Property	An individual piece of data attached to a Biskit[446]. A property will have a name, a type and a label as well as other meta properties[448] defining how it is organised.
Property Path	When adding a property[448] to a report, or placing a condition[447] on a property[448], you can choose a property[448]

© 2010-2015 Exprodo Software

	of a property[448]. This is known as a property path. For example, if you want a Permission[448] to prevent creating reports that are scheduled to run daily, you would need a condition[447] on the the report schedule's repeat type. This is found in the property path **reportSchedule.repeat.repeatType**.
References	Biskit[446] properties[448] may have various types, such as integer and string. One such type is Biskit[446], which means that a Biskit[446] may reference another Biskit[446]. The References for a Biskit[446] is the list of all such referencing Biskits[446].
Relation	Provides a means of comparing the value of a property[448] to another value e.g. equals, less than etc.
Repeat	Something that occurs at regular time intervals.
Requested Booking	A booking[446] that has been created but not yet approved[446].
Requested Project	A project[448] that has been created but not yet approved[446].
Requested User	A user that has been created but not yet approved[446].
Resource	A room, person, instrument, or anything else that that must be booked before it can be used.
Resource Group	A way of grouping similar resources[449] together. A resource[449] may belong to many groups. See also Resource Type[449].
Resource Type	A way of grouping similar resources[449] together. A resource[449] has only one type. See also Resource Group[449].
Resource Usage	The amount of time a resource[449] is actually used as opposed to the time it is booked[446] for.
Resource Usage Calendar	A page which provides a graphical view of resource usage[449] by day, week or month.
Resource Usage Recorder	A page that allows you to indicate when a resource[449] is actually used. Also known as an Electronic Log.
Settings	See User Settings[450].
Service Order	A request for a Service[449], an Order will have a status and a creator
Service Provider	The provider of a Service[449]. A provider may provide many different Services[449].
Services	The ability to set up different tasks that a user may request
Single Item Report	Shows a report of a single item of the appropriate Biskit Type[446].
Summary Report	A report like a spreadsheet pivot table that can automatically count or sum data.
System Event	Various things that happen within **Calpendo** are recorded as a System Event. For example, a system event is recorded each time an email is sent or an error is detected.
System Usage Statistics	These are statistics that may optionally be kept to track how much each user has been using **Calpendo**.
Task	An **Exprodo SDM** concept of an individual action that needs to be performed. See also Event.[447]
Template	See Time Template[450] or Project Template[448].
Time Template	Allows the administrator to assign an acceptability rating to bookings[446] depending on when they are booked, who

© 2010-2015 Exprodo Software

	makes the booking and the project[448] the booking is for.
Time Templates Calendar	A page which provides a graphical view of Time Templates[450] by day, week or month.
Units	A method of displaying different measurement types such as feet and metres.
User Approval Process	The process by which new users become approved.
User Group	A way of grouping similar users together. A user may belong to many groups. See also User Type[450].
User Project Association	A list of projects[448] the user is associated with.
User Roles	Users may be assigned roles, and then Permissions[448] can be configured to use those Rules. For example, by default, users with the **Admin** role[450] can modify more things than users without the **Admin** role.
User Settings	Specifies settings that change the way **Calpendo** operates for this user. These will override the Global Preferences[447].
User Status	The current status of the users account. This may be one of Requested, Normal, Password must be reset at next login, Blocked or Denied.
User Type	A way of grouping similar users together. A user has one type. See also User Group[450].

© 2010-2015 Exprodo Software

Index

- A -

Abdicate 58
Actual Resource Usage
  Enabling For A Particular Resource 273
Admin Role
  Introduction To 150
  Projects That Can Be Booked 144
  Receiving New User Request Emails 201
Advanced Booking Rule
  API 258
Appearance
  Global Preferences For 329
Approving
  Booking Requests 139
  New User Requests 153
  Project Requests 145
Attachment, File
  Attaching Documrents to Biskits 204
  Creating A Set Of Bisket Def 398
  Example Adding Properties For 390
Authentication Methods 322
  Basic 323
  Editor 324
  External 323
  IMAP 322
  Local 322
  SMTP 322
Automatic Email 297
  Conditions 300
  Dealing With Sensitive Data 305
  Details 299
  Editor 298
  Examples 306
  Property Path 301
  Recipients 300
  Subject Tab 302
  Tree 298
  URL 304

- B -

Backing Up The Database 430
  Use Custom Per-Table Settings 431
Bakery 351
  Biskit Definitions 353
  Biskit Format 356
  Biskit Inheritance 357
  Biskit Property Definitions 363
  Bit Sets 368
  Boolean Property Definitions 364
  Check Reserved Words Preference 339
  Double Property Definitions 365
  Edit Mode 380
  Editor 378
  Editor Example 384
  Example Adding A New Yes-No Mapped Int Property 394
  Example Adding Created/Updated/Version Properties 392
  Example Adding File Attachment Properties 390
  Example Adding Formulaic Properties 387
  Example Adding Properties 384
  Example Booking Cost Formulae 387
  Example Creating A Hierarchy Of Bisket Def 402
  Example Creating A Master Slave Biskit Relationship 407
  Example Creating A New Basic Bisket Def 397
  Example Creating A Set Of Bisket Def 398
  Example Creating An Inheriting Biskit Def (Booking) 412
  Formulae 375
  Formulaic Properties 375
  Integer Property Definitions 366
  Integer Type 366
  Java Enum 369
  Java Enum Definitions 369
  Java Enum Property Definitions 369
  Mapped Integers 366
  Mapped Strings 373
  Master Slave 363
  Property Definitions 358
  Property Layout Editor 413
  Property Storage Mechanisms 351
  Run Data Definition Validation At Boot 339
  Set And List Property Definitions 370

© 2010-2015 Exprodo Software

Bakery 351
String Enum 374
String Enum Property Definitions 374
String Enumerations 375
String Property Definitions 372
Updating The DB 378
User Roles 368
View Mode 379
Banner, Global Preferences For 329
BeanShell
Advanced Booking Rules 256
Biskit
Biskit Property Definitions 363
Biskit Property Type 363, 371
Collection Editor, Properties Visible In 132, 361
Definitions 353
Detail, Properties Visible In 132, 361
Dynamic Biskits 356
Format 356
Inheritance 357
Integer Property Definitions 366
Integer Type 366
Java Enum 369
Java Enum Property Definitions 369
List, Properties Visible In 132, 361
Many To Many Properties 371
Master Slave 363
Properties Visible In Biskit Detail 132, 361
Properties Visible In Collection Of 132, 361
Properties Visible In List Of 132, 361
Run Data Definition Validation At Boot 339
Set And List Property Definitions 370
String Enum 374
String Enum Property Definitions 374
String Property Definitions 372
String Property Type 372
String Type 373
Biskit Property Type 363, 371
Bit Sets 368
Booking Rules 60
Advanced Rule API 258
Advanced Rules 256
Applies To 226
Apply To Admin, Preference For 345
Applying To Bookings 226
Booking Time To Include 254
Bookings To Include In Count 253
Choosing Between Time Templates, Booking Rules And Permissions 212
Configuring 223
Does Not Apply To 229
Double Booking Rules 236
Duration Rule 243
Editor 224
Exclusive Booking Rules 236
Global Preferences For 345
Holiday Rule 244
How They Work 223
Interval Rule 245
Java Rules 256
Number Of Bookings Rule 246
Predefined Slots Rule 249
Pre-requisite Booking Rules 236
Search Results Rule 250
Simple 230
Total Time Booked Rule 251
Types 229
Validator 264
Booking Sub Type
Example Creating An Inheriting Biskit Def 412
Setting Up 203
Bookings 36
Administration 136
Approval Process 138
Approving 139
Automatic Booking Approval Process 210
Background Colours 331
Background Tooltips 331
Best Possible 50
Bookmarks 38
Calendar 36
Calendar, Creating Bookings 49
Calendar, Display 47
Calendar, Editing and Cancelling Bookings 54
Calendar, Exploring 37
Canceling 56
Changing More Than The Status 141
Colours, User Settings For 33
Controlling Bookings 210
Creating Bookings In The Past 274
Creating Repeat Bookings 53
Day, Week, Month View 41
Default Status 331
Default Status For Bookings Created By Admin 331

© 2010-2015 Exprodo Software

Bookings 36
Default Status For Bookings Created By Users 331
Deleting 56
Enabling Changes To Old Bookings 274
Enabling Pre-defined Time Slots 275
Example Creating An Inheriting Biskit Def 412
Filtering On The Calendar 40
Format 331
My Bookings 66
My Projects' Bookings 67
Owner 331
Permissions 60
Properties 137
Reminder For A Particular Booking 52
Reminder, Global Preferences For 331
Reminders, User Settings For 32
Repeat Flushing, Time Between 339
Requests Page 139
Resource 38
Resource Columns 331
Restrictions 57
Search 68
Search For Cancellations 65
Searching For 61
Status, Default of 50
Status, Filter For The Calendar 40
Sub Type 203
Template Selection 39
Time Display 331
Time Templates 57
Tooltips, User Settings For 33
Using Searches 61
Bookings To Count
Double Booking Rule Advanced Settings 238
Bookings To Ignore
Double Booking Rule Advanced Settings 238
Bookmark Manager
User Permissions 169
Bookmarks
Bookings 38
Calendar 168
Editing 171
Manager 168
User Created 168
Boolean
Boolean Property Definitions 364

- C -

Calendar
Bookings 36
Bookings, Creating 49
Bookings, Display 47
Bookings, Editing and Cancelling 54
Bookings, Exploring 37
Exporting iCal 45
iCal Export 45
Printable View 46
Resource Usage 82
Time Templates 220
Calpendo Administration Guide 136
Calpendo Configuration Guide 198
Calpendo Quick Start Guide 194
Calpendo User Guide 28
Changes for 7.0
Administrators 13
Changes for 7.0.1-45 17
Administrators 18
Regular Users 17
Changes for 7.0.46-55 23
Administrators 23
Regular Users 23
Changes for7.0
Regular Users 12
Conditions
Advanced 98
Anatomy Of A Condition 98
Automatic Email 300
Combination Options 105
Context 99
Date Properties And Accuracy 104
Nested Conditions 106
Panel 99
Pattern Matching Text 102
Property Path 101
Referenced By Property Path 107
Relation 102
Text Pattern Matching 102
Type 100
Value 104
Configuring
Booking Rules 223
Email Preferences 201

© 2010-2015 Exprodo Software

Configuring
Groups 266
Initial 201
Location 270
New User Requests 201
Project Properties 206
Project Template 206
Projects 205
Questions 198
Resources 201, 270
Service Orders 276
Service Providers 276
Services 276
Time Templates 215
Types 266
Converting From
Existing Booking System 205
Creating Users 162
CSS, Preference Setting For 329

- D -

Data Explorer 127
Editing Multiple Items 128
How To Edit A Single Item 130
Database
Backing Up 430
Mapping Your Own Tables 356
Database Dumps
Minimum Time Between 339
Date And Time Format
Global Preferences For 337
Double
Double Property Definitions 365
Double Booking Rule
Advanced Settings 238
Bookings To Count 238
Bookings To Ignore 238
Handling Holidays 241
Dynamic
Biskits 356

- E -

Editor
Authentication Methods 324
Automatic Email 298
Bakery 378
Booking Rules 224
Bookmarks 168
FAQ 325
Import 184
Location 270
Mapped Int 382
Mapped String 382
Menu 311
Permissions 285
Property Layout 413
Reports 121
Resource Usage Outcome 269
Resources 270
Service Providers 276
Services 276
String Enum Def 382
Time Templates 217
Types And Groups 266
Units 383
Electronic Log (See Resource Usage) 79
Email
Apparent Sender 338
Authentication 338
Automatic Email 297
Connection Security 338
Copy All Outgoing Emails 338
Global Preferences For 338
Initial Configuration 201
Limiting Who Can Be Sent Emails 338
Reply To 338
Sending Enabled, Global Preference For 338
Signature On Outgoing Email 338
SMTP Server 338
URLs Refering To Calpendo, Global Preference For 338
Using Calpendo To Send 172
Example Automatic Email
Dealing With Sensitive Data 305
Send An Email When A Booking Is Cancelled 309
Send Email To A User Whenever Thier New User Request Is Denied 308
Send Email To An Adminitrator When A Project Request Created 306
Example Bakery
Adding A New Yes-No Mapped Int Property 394
Adding Created/Updated/Version Properties 392

© 2010-2015 Exprodo Software

Example Bakery
Adding File Attachment Properties 390
Adding Formulaic Properties 387
Adding Properties 384
Booking Cost Formulae 387
Creating A Hierarchy Of Bisket Def 402
Creating A Master Slave Biskit Relationship 407
Creating A New Basic Bisket Def 397
Creating A Set Of Bisket Def 398
Creating An Inheriting Biskit Def (Booking) 412
Example Booking Rule
Disallowing Bookings More Than 6 Weeks In Advance 231
Handling Holidays 241
Number of Bookings Rule Example 247
Physicist Projects Can Book 4 Weeks In Advance, Biologists 6 Weeks 232
Prevent Cancellations With Less Than 24 Hours To Go 235
Special Projects And Admins Can Book Any Time In Advance 234
Special Projects Can Book Any Time In Advance 233
Total Time Booked Rule Example 255
Example Booking Rule (Advanced)
Enforce 15 Minute Gaps Between Bookings 262
Enforcing Project Resource Settings 260
Reject Bookings More Than 6 Weeks Into The Future 259
Example Permission
Admins May Create Or Update Anything 291
Anbody Can Modify A Booking Request They Created If The Booking Is Still A Request 290
Anybody May Create A Booking Request 289
Approval Of A Booking Requires An Admin 294
Configuring Multi-Step Approval Process 208
Hiding Resources And Booking Properties 296
No One Can Email Password Details 295
Only Allow Specified Users To See The Price Being Charged For A Resource 292
Existing Booking System
Converting From 205
Explorer, Data 127
Exprodo 351

- F -

FAQ 325
Configuring FAQ Answers 326
Viewing 133
Far Term
Time Template 60
File, Attachment
Attaching Documrents to Biskits 204
Creating A Set Of Bisket Def 398
Example Adding Properties For 390
Formulae
Known Problems 377
Frequently Asked Questions
FAQ Editor 325
Viewing 133

- G -

General
Global Preferences For 339
Getting Started 10
Global Preferences 328
Appearance 329
Booking Calendar Background Template Colours 332
Booking Calendar Date Format 334
Booking Calendar Start and Finish Time 335
Booking Calendar Tooltip When Template Has No Message 333
Booking Reminders 331
Booking Resource Columns 331
Booking Rules 345
Booking Status, Default Admin 331
Booking Status, Default User 331
Bookings 331
Bookings Miscellaneous Settings 335
Check Reserved Words 339
Copy All Outgoing Emails 338
CSS 329
Database Dumps, Minimum Time Between 339
Date And Time Format 337
Default Booking Status 331
Default Initial Page 340
Email 338
Email Authentication 338
Email Connection Security 338

© 2010-2015 Exprodo Software

Global Preferences 328
Email Signature 338
Email SMTP Server 338
Email, Apparent Sender 338
Email, Limiting Who Emails Can Be Sent To 338
Email, Reply To 338
Email, Sending Enabled 338
Email, URLs Refering To Calpendo 338
General 339
Hacking Protection 347
Host Page Title 330
Icon 339
Licence 340
Login Header 330
Login Name Length, Minimum 350
Menu 341
Menu, Default Admin 341
Menu, Default Root 341
Menu, Default User 341
Network Metrics 342
Network Metrics, Enabled, Preference For 342
Network Metrics, Minutes Between Sending 342
Network Metrics, Network Calls Between Sending 342
New User Default Roles 350
Page Banner 329, 330
Page Banner In Debug Mode 329
Password Content Requirements 346
Passwords, Allowing Browser To Remember 346
Project Rendering Type 343
Project Request Filtering 343
Project, Default Name 343
Project, Template 343
Projects 343
Read-only Mode 339
Repeat Flushing, Time Between 339
Request Filtering By User Type 350
Resource Usage 344
Resource Usage Future Calendar Colour 344
Resource Usage Good Outcome 344
Rules, Apply To Admin 345
Run Data Definition Validation At Boot 339
Security 346
Submit Crash Reports 339
System Usage 348
Template Project 343
Templates Apply To Admin 349
Temporary Directory 339
Time Templates 349
Usage Statistics Enabled 348
Usage Statistics, Time Between Flushes 348
Usage Statistics, Time Per Statistic 348
User Session Timeout 350
User Type, Requirement For 350
Users 350
Glossary 446
Group Report 115
Groups
Project Groups 265
Resource Groups 265
User Groups 265

- H -

History
History Button 97
History Page 126
Holidays
Comparing Holiday Rule, Time Templates And Double Booking Rule Holiday Handling 242
Displaying Warning With Double Booking Rule 241
Holiday Rule 244

- I -

Import 177
Attachments 187
Data Errors 189
Example Bookings File 182
Example Files 177
Example Project File 180
Example Resource File 179
Example Time Template File 181
Example User File 178
File Attachments 187
Handling Errors 187
Header Errors 188
Import Editor 184
Options 186
The Import Page 184
Violation Errors 190
Initial Configuration
Before Going Live 201

© 2010-2015 Exprodo Software

Initial Configuration
Email Preferences 201
New User Requests 201
Resources 201
Integer Type
Property 366

- J -

Java
Advanced Booking Rules 256
Enum Definition 369
Enumerations 369
Java Enum
Property 369

- L -

Layout Editor
CSS Rule Editors 428
Edit Mode 420
Information Pane 418
Moving Properties 421
Notes 425
Organisation Pane 417
Preview 416
Property Group Header Information 418
Property Table Types 422
Licence
Global Preferences For 340
List Report 110
Location
Configuring 270
Location Editor 270
Login Name
Minimum Length 350

- M -

Mapped Integers 366
Mapped Strings 373
Menu
Default Admin Menu 341
Default Root Menu 341
Default User Menu 341
Global Preferences For 341
Menu Editor 311
Menu Editor 311
Biskit Tree Viewer 317
Creating a New Menu 319
Custom Search Descriptors 317
Custom Search Pages 317
More Calpendo Configuration
Booking Sub Types 203
File Attachments 203
Training Records 203
My Bookings 66
My Projects 76
My Projects' Bookings 67
My Projects' Orders 95
My Projects' Resource Usage 88

- N -

Near Term
Time Template 60
Network Metrics
Enabled, Preference For 342
Global Preferences For 342
Minutes Between Sending, Preference For 342
Network Calls Between Sending, Preference For 342

- P -

Page Banner, Global Preferences For 329
Password
Allowing Browser To Remember 346
Change 31
Requirements, Preference Setting For 346
Permission Denied
Choosing Between Time Templates, Rules And Permissions 213
Permission Denied Message 213
Permissions
Actions 283
Applies To 288
Breaking An Activity Into Its Parts 279
Choosing Between Time Templates, Booking Rules And Permissions 212
Choosing Between Time Templates, Rules And Permissions 213
Conditions 281, 287
Details 286
Does Not Apply To 289

© 2010-2015 Exprodo Software

Permissions
Examples 289
How Permissions Work 278
Layering 279
Overview 278
Permission Denied Message 213
Permissions Editor 285
Positive and Negative 281
Precedence Of 280
Targeting Users 282
Tree 286
Preferences
Global 328
Project Association 78
Projects 69
Administration 143
Approval Process 145
Approval, Multi-step Process 146
Approval, Single-step Process 145
Approving Or Denying 147
Changing More Than The Status 148
Changing Users Associated With 164
Configuring 205
Configuring Approval Process 208
Configuring Multi-Step Approval Process 208
Configuring Single-Step Approval Process 208
Creating 70
Default Name, Preference Setting For 343
Global Preferences For 343
Modifiying 72
My Projects 76
Properties 144, 206
Rendering Type, Preference Setting For 343
Requests Page 146
Requests, Filtering By User Type, Preference Setting For 343
Search 77
Searching 72
Template Configuration 206
Template, Preference Setting For 343
User Association Request 78
Using Search 72
Property
Biskit Property Definitions 363
BiskitPropertyType 363, 371
Boolean Property Definitions 364
Double Property Definitions 365
Integer Property Definitions 366
Integer Type 366
Java Enum 369
Java Enum Property Definitions 369
Layout Editor 413
Null Allowed 362
Project 362
Property Definitions 358
Required Fields 362
Set And List Property Definitions 370
String Enum 374
String Enum Property Definitions 374
String Property Definitions 372
String Property Type 372
String Type 373
Tab Layout 362
Visibility When Biskit Detail Viewed 132, 361
Visibility When Biskit List Viewed 132, 361

- R -

Read-only Mode
Consequences Of Putting Into Read-Only Mode 433
Creating A Read-only Copy 432
Global Preference For 339
Requirements For A Read-Only Database 432
References 97
Reminders
For A Particular Booking 52
Global Booking Reminder Preferences 331
Repeat
Bookings, Creating 53
Time Between Repeat Flushing 339
Reports
Manager 176
Report Editor 121
Scheduling 124
Request
Project 146
Project Association 78
User 153
Resource Usage 79
Calendar 82
Calendar Future Background Colour, Preference Setting For 344
During Usage Page 81
Enabling For A Particular Resource 273
Global Preferences For 344

© 2010-2015 Exprodo Software

Resource Usage 79
IP Address Restriction, Resource Setting For 273
Large Font, Resource Setting For 273
My Resource Usage 87
Outcome When All Went Well, Preference Setting For 344
Pre-Usage Page 80
Recorder, Session 79
Resource Selector Page 79
Resource User Selectable 273
Search 84, 89
Search Project Resource Usage 88
Session ID Template, Resource Setting For 273
Session Recorder 79
Using Search 84
Resource Usage Outcome
Editor 269
Resources
Configuring 270
Initial Configuration 201
Resource Editor 270
Selection 44
Usage Recorder, IP Address Restriction 273
Usage Recorder, Large Font Settings 273
Usage Recorder, Session ID Template 273
Roles
Admin 150
Admin And Projects That Can Be Booked 144
Description Of 150
Receiving New User Request Emails 201
Root 150
User Role, Editing Role Definition 368
Root Role
Introduction To 150

- S -

Search
Booking 68
Booking Cancellations 65
Comples Content Types 113
Editing Search Information 117
Exporting Information 120
General Search 108
Group Report 115
List Report 110
My Bookings 66
My Orders 93
My Projects 76
My Projects' Bookings 67
My Resource Usage 87
Order Search 91
Project 77
Project Orders 95
Project Resource Usage 88
Resource Usage 84, 89
Resource Usage Search 84
Saving And Reusing 119
Service Orders 91
Summary Report 112
Using Bookings 61
Using Project 72
Searching 108
Security
Brutce Force Password Hacking, Preference Setting For 347
Global Preferences For 346
Password Requirements, Preference Setting For 346
Password, Allowing Browser To Remember, Preference Setting For 346
Service Orders
My Orders 93
Search 91
Search Project Orders 95
Using Search 91
Service Providers
Configuring 276
Service Provider Editor 276
Services 90
Available Services 90
Configuring 276
Service Editor 276
Service Orders 276
Setting Up Different Booking Sub Types For Resources 203
Single Sign On 322
Starting
Quick Start Guide 194
Statistics
Network Metrics, Enabled, Preference For 342
Network Metrics, Minutes Between Sending, Preference For 342

© 2010-2015 Exprodo Software

Statistics
Network Metrics, Network Calls Between Sending, Preference For 342
Usage Statistics Enabled, Preference For 348
Usage Statistics Time Between Flushes, Preference For 348
Usage Statistics, Time Per Statistic, Preference For 348
Status
Booking Filter For The Calendar 40
String Enum
Property 374
String Enumerations 375
String Property Type 372
String Type 373
Summary Report 112
Comples Content Types 113
System Events 191
System Usage
Global Preferences For 348

- T -

Template
Project 206
Text
Pattern Matching in Conditions 102
Time Booked Rule
Limiting 251
Time Templates
Abdicate 58
Apply To Admin, Preference For 349
Bookings 57
Calendar 220
Choosing Between Time Templates, Booking Rules And Permissions 212
Configuring 215
Editing 223
Editor 217
Far Term 60
How Time Templates Work 215
Near Term 60
Preferences For 349
Putting On The Calendar 222
Time Template Preferences 349
Toolbar Button
Standard Definitions 96
Tooltips
Booking, User Settings For 33
Total Time Booked Rule
Booking Time To Include 254
Bookings To Include In Count 253
Training
Checking For Before Booking 203
Types
Booking Types 265
Project Types 265
Resource Types 265
User Types 265
Types and Groups
Configuring 266
Description 265
Editor 266

- U -

URL
Refering To Calpendo In Emails 338
Within Automatic Email 304
Usage Statistics
Enabled, Preference For 348
Time Between Flushes, Preference For 348
Time Per Statistic, Preference For 348
User
Administration 150
Approving New Users 153
Approving Or Denying 154
Changing Group Membership 157
Changing More Than The Status 155
Creating An Account 28
Creating Users 162
Global Preferences For 350
Initial Configuration Of New User Requests 201
Modifying Users 163
New User Default Roles 350
Password Reset 163
Project Associations, Changing 164
Properties Of A User 150
Requests Page 153
Requirement For User Type 350
Reset Password 163
Roles 368
Search 158
Session Timeout 350
Settings 32, 167
Special 167

© 2010-2015 Exprodo Software

User
User Group Membership, Changing 165
User Group
User Membership, Changing 165
User Requests Page
Request Filtering By User Type 350
User Settings
Booking Reminders 32
Calendar View 33
Calpendo 32
Date & Time 35
Editing For Another User 167
Editing Your Own Settings 32
Email 35
Menu 35
User Type
Requirement For 350

- V -

Version History 12
Visibility
Of Properties In A Biskit Collection Editor 132, 361
Of Properties In A Biskit List 132, 361
Of Properties In Biskit Detail 132, 361

- W -

Web Browser Compatability 133

© 2010-2015 Exprodo Software

Calpendo is produced by Exprodo Software
www.exprodo.com
Email: calpendo@exprodo.com
UK Telephone: 01235 813458
International Telephone: +44 1235 813458